MPSYS4

Microprobe Data Analysis System

User's Manual

Microanalytical Research Centre
School of Physics
University of Melbourne
VICTORIA 3010
AUSTRALIA
Fax: + 61 (0)3 9347 4783
Ph: + 61 (0)3 8344 5376
Email: marc@physics.unimelb.edu.au
Web: http://www.ph.unimelb.edu.au/marco

January 2002

Preface

This manual contains an alphabetical listing of help files for all the commands and other command structures in the MpSys data collection and manipulation package. Information is provided on syntax for commands as well as detailed information on the Skip programming language incorporated with MpSys. Please read the MpSys User's Manual for further information.

For further technical assistance please contact MARC via the following email address:

marc@physics.unimelb.edu.au

Limitation of Liability

Micro Analytical Research Centre does not assume any liability arising out of the use of the information contained within this manual. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Micro Analytical Research Centre, nor the rights of others.

Micro Analytical Research Centre will not be liable for any defect in hardware or software or loss or inadequacy of data of any kind, or for any direct, indirect, incidental, or consequential damages in connections with or arising out of the performance or use of any of its products. The foregoing limitation of liability shall be equally applicable to any service provided by Micro Analytical Research Centre.

Note

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying or otherwise, without the prior written permission of MARC.

Manual Version: 1.0

Manual Date: January 2002

©2002 Microanalytical Research Centre

Table of Contents

1. INTRODUCTION 5

2. STARTING MPSYS 8

ON-LINE HELP 8

3. MPSYS QUICK START 9

THE FUNCTION BUTTONS 9
EXAMPLE DESK TOP 11
STARTING A RUN 12
SETTING THE EXPERIMENTAL PARAMETERS IN THE JOURNAL FILE 13
WORKING WITH A SPECTRUM IN A SPECTRUM WINDOW 23
Spectrum control buttons 23
Calibrating an energy spectrum 24
Element labels 27
Using the element selection button 28
Resizing spectra 29
Unknown element identification 31
GENERATING A TUNED DISPLAY OR MAP FROM WINDOWS SET IN A SPECTRUM 32
EXTRACTING SPECTRA FROM MAPS 34
THE MAIN MENU BAR 37

4. WORKING WITH MAP WINDOWS 39

LOADING MAPS 39
SAVING MAPS 39
PRINTING MAPS 39
CLEARING MAPS 40
CLOSING MAP WINDOWS 40
MANIPULATING MAP COLOURS 40

5. JOURNAL FILE OPTIONS 42

General Parameters 43
Experiment Parameters 43
Station Parameters 48

6. THE SKIP LANGUAGE 52

COMMAND ALIASING 52
MACRO EXECUTION 52
CALCULATOR 53
INPUT AND OUTPUT REDIRECTION 53
ONLINE HELP 54
A PROGRAMMING LANGUAGE 54

7. BRIEF COMMAND SUMMARY 56

GENERAL COMMANDS 56
WINDOW COMMANDS 56
DATA COMMANDS 57
DATA ACQUISITION 57
SPECTRA COMMANDS 58
MAP COMMANDS 59
FILE SYSTEM COMMANDS 59

8. JOURNAL PARAMETER SUMMARY 60

GENERAL PARAMETERS 60

EXPERIMENT PARAMETERS 60
General *60*
Beam line *61*
Beam *61*
Chamber *61*
Scanning *62*
Deadtime *62*
STATION PARAMETERS 63
General *63*

9. FILE FORMATS 64

MPSYS RAW EVENT DATA (.EVT) 64
MPSYS SORTED EVENT DATA (.SD, .SP) 66
MPSYS SPECTRA FILES (.IMG) 67
MPSYS MAP FILES (.MAP) 68

10. MPSYS X-RAY LINE DATABASE 69

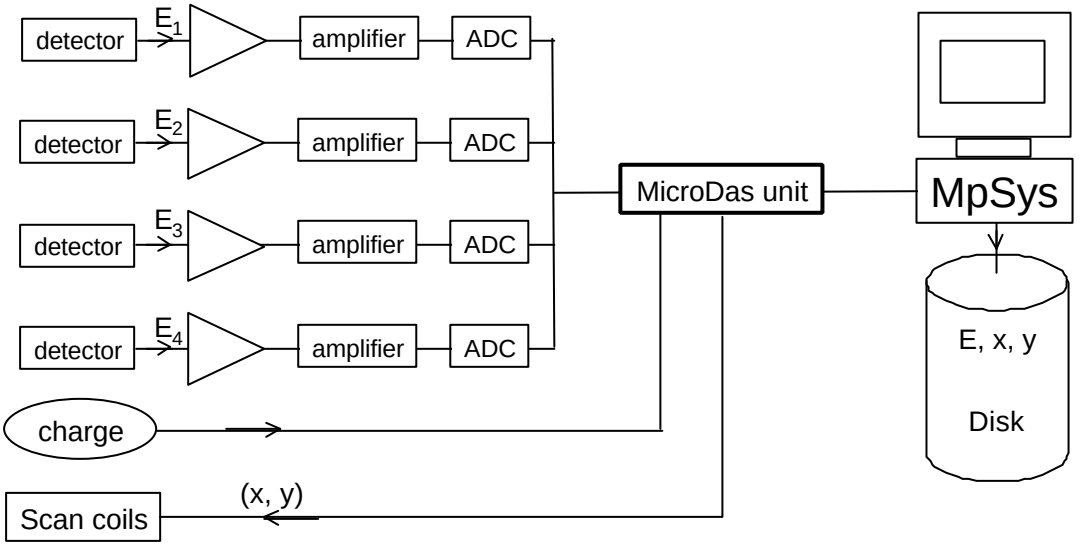
11. WORKING WITH MPSYS IN A MICROSOFT WINDOWS ENVIRONMENT 70

1. Introduction

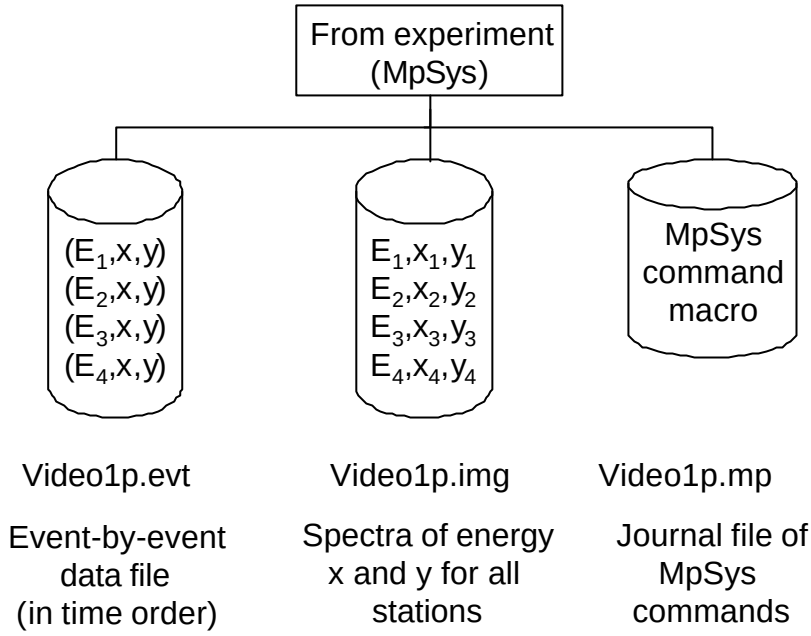
MpSys is control program primarily designed for nuclear microscopy. It has the following features:

- Data collection from up to four "stations". A station usually has a one of the detectors typically employed with a nuclear microprobe connected to it such as an x-ray detector or a detector of backscattered particles.
- A scan signal generator for control of the scan of the beam spot over the region of interest on the specimen.
- Provision for tagging each event received in any of the four stations with the corresponding scan coordinates and storing the event in time sequence on disk. This is event-by-event mode of data acquisition.
- Extensive features for manipulation and display of the incoming data stream. Most commonly this consists of the display of the energy spectra from each of the four stations, together with "tuned displays". These are intensity maps of the incoming data derived from windows placed in the energy spectra of the four stations. One or more tuned displays may be derived from one or more windows placed in each energy spectrum. Windows may also be placed in the x- and y-spectra.
- Extensive features for manipulation and display of the data off-line. This includes sorting the data into energy and position order for rapid extraction of maps. A map is the off-line equivalent of a tuned display.
- MpSys collects data into the event-by-event file for the duration of a "run". The user determines the duration of a run, which is typically for a time long enough to collect statistically significant spectra from the region of interest within the scan.

The diagram below shows the typical experimental set-up for the use of MpSys to collect data. The energy signals from any of the four detectors (E_1, E_2, E_3, E_4) is interfaced to MpSys via a MicroDas unit and a data acquisition card in the linux workstation.



Data acquisition system with MpSys

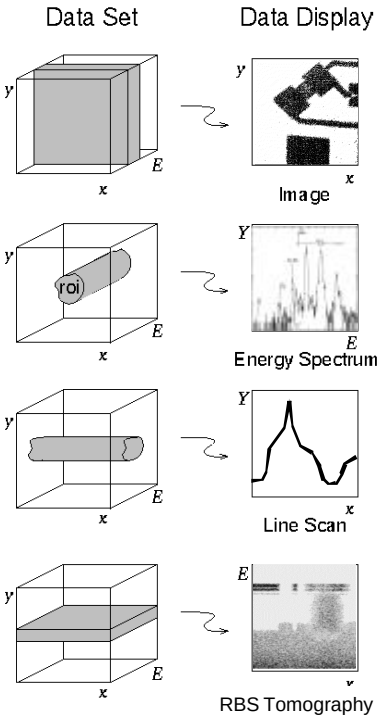


Files created by MpSys following a run with example file names for run "videop1".

At the conclusion of a run, MpSys closes the files associated with the run. As shown in the above diagram, these are the event-by-event data file (extension .evt), the energy, x- and y-spectra associated with each station concatenated into one file (extension .img) and the MpSys command macro (extension .mp) which holds all the experimental parameters associated with the run. This macro is used to reconfigure MpSys to the parameters associated with the run in the future.

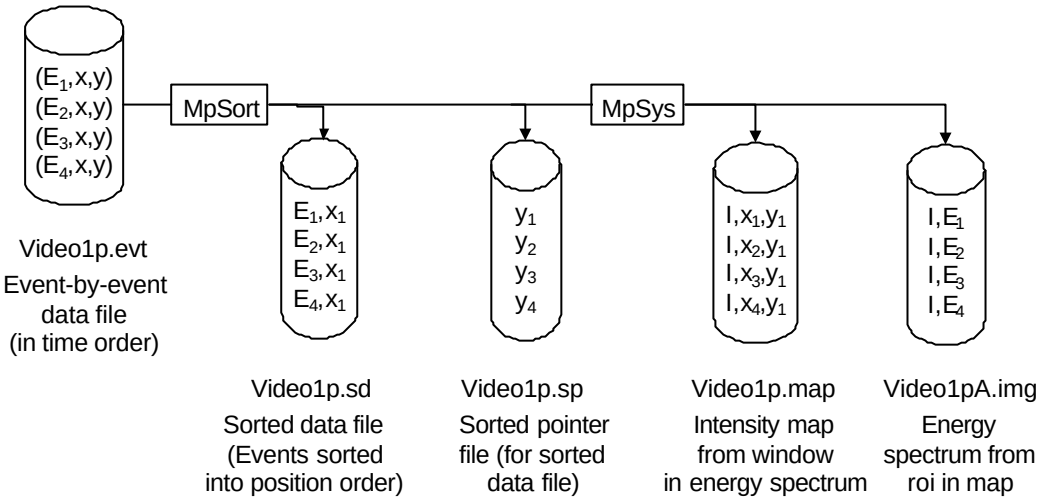
This combination of event-by-event data and the MpSys command macro means that the entire experiment can be replayed off-line and no data associated with the experiment is discarded.

After the run has concluded, the event-by-event file may be sorted and the data redisplayed in a variety of formats. Some of the possibilities are illustrated in the following diagram.



The typical steps tp data processing wth MpSys is illustrated in the following diagram. The steps may be summarised as:

1. Run MpSys and collect data from an experiment. Data collected into an event-by-event file (.evt) and an associated file of containing the energy, x and y spectra from each station (.img).
2. Sort the event-by-event file to produce a sorted data file (.sd) and its associated sorted pointers file (.sp).
3. Load the sorted data file into MpSys and create maps from energy windows set in the energy spectrum (.map).
4. Draw shapes in a map to define a region of interest (roi) on the sample. Extract from the shape the energy spectrum for the region of interest.
5. Further processing of the energy spectra from roi's with other software packages (e.g. nufit, RUMP, GUPIXE, etc).



Each step in this process is described in more detail in the following sections of this manual.

2. Starting MpSys

MpSys is started from the command line prompt of a Unix X-window. On some systems an icon on the desktop may be used to invoke MpSys. Several arguments may be added to the command to modify the characteristics of MpSys when it starts. Most users will simply type **mpsys** at the Unix command line:

```
praxis:dnj/ldata/dnj/run1$ mpsys
```

Below is the full command line specification for running MpSys:

mpsys [login_macro][-display dname][-fontc name1][-fontl name2][-fonte name3][-font name4] [-fg fcolor][-bg bcolor][-bd bdcolor][-geom geom]

The MpSys macro file login_macro will be executed first if it exists. It will be searched for in $HOME/macros. If a macro is not specified, the default login macro login.mp will be executed if it is found.

dname	is the name of the X Windows display address you want to use for displaying all MpSys windows. An example address could be **marc@physics.unimelb.edu.au:0.0**
name1	commands text font name
name2	labels text font name
name3	exponents font name
name4	general text font name
fcolor	foreground colour of X Windows
bcolor	background colour of X Windows
bdcolor	border colour of X Windows
geom	allows you to specify initial size and position of commands window where geometry is given as width x height + x + y in pixels

On-line Help

The help files are also available on-line while using MpSys and are accessed by typing:

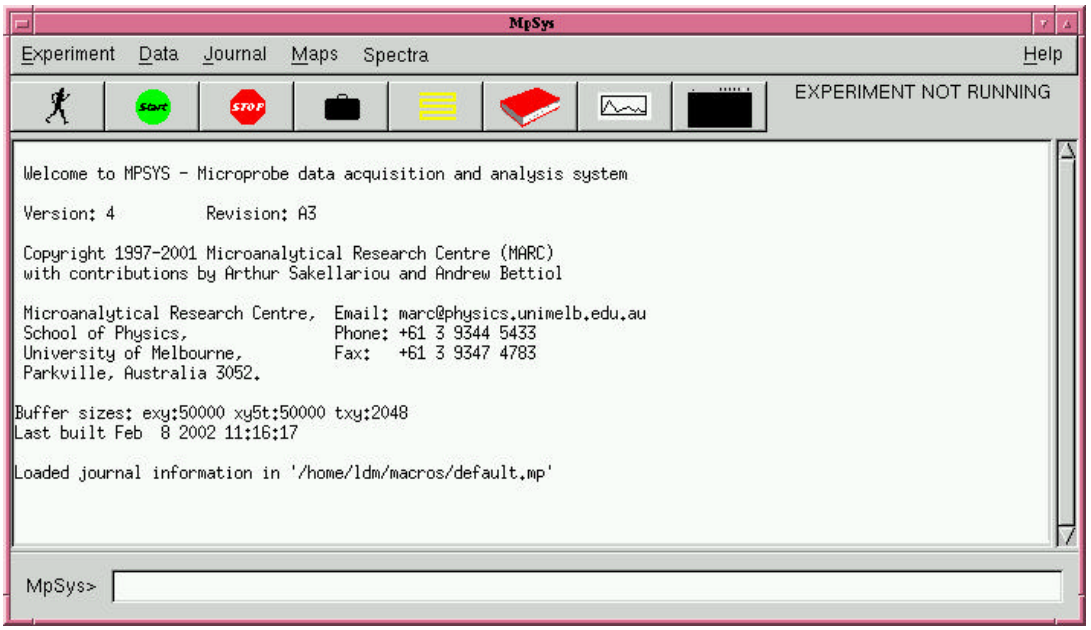
help <topic>

Where <topic> is the capitalised topic name in the following listing. Please remember, however, to type the topic name in lower case letters to be compatible with MpSys and the UNIX operating system under which MpSys runs.

3. MpSys Quick Start

This section provides a brief introduction to MpSys and describes how to perform the most common application: the collection and display of images and spectra.

After starting MpSys from the unix command prompt, the main MpSys window appears.



This window gives quick access to all of the most frequently used MpSys functions. The window may be resized by clicking and dragging on the corners.

MpSys can be controlled by two methods using this window. The most convenient method is by use of the function buttons and the pull down menus from the task bar. Advanced functions can be accessed from the command prompt window at the bottom of this window. This provides access to the full power of the Skip command language embedded within MpSys and allows users to run macros of stored MpSys commands.

The function buttons

Reading from left to right, the function buttons are arranged the same sequence used to start a run.

The ***run*** button prepares MpSys for a new run. All data areas are cleared and the system is initialized ready to start the collection of data which will be stored in a new file. At the same time, the scanning menu appears to confirm the file name for the new data and allow any last minute changes to the scan parameters. Note that the scan parameters cannot be changed after the run has started. Once these parameters have been set, the run is ready to ***start***.

The ***start*** button causes the data collection to start. The scan will commence and data will start appearing in windows. It is now not possible to change the scan parameters, although they can be reviewed at any time during the run by pushing the ***scan*** button. The run may be stopped and started at any time, with all data going into the same file. Note that the scan will recommence from the start each time the ***start*** button is pressed.

The ***stop*** button causes the data collection to stop, along with the scan, but does not close the file being used to store the new data. This means that the run can be restarted by pressing the ***start*** button.

The ***close*** button is used at the conclusion of the run to close the file used to store the data and stop the data collection process. The system will now be ready for a new run.

The ***scan*** button allows access to the scan menu. If there is no run in progress, this menu can be used to change the scan parameters. If there is a run in progress, the apply button is missing and the menu may only be used to review the current scan parameters. The scan menu is discussed in more detail below.

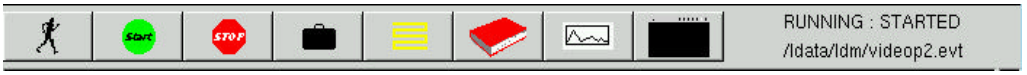
The journal button allows access to the journal file that holds all of the experimental parameters of the run. This includes the beam energy and particle, details about the present specimen and the operating parameters of the system hardware for future reference. Some of these experimental parameters affect other MpSys functions, particularly those that deal with elemental identification and energy calibration and so must be set correctly at the start of a run for these functions to perform correctly.

The ***new spectrum*** button creates a new window on the screen for display of a spectrum. The spectrum displayed in the window may be selected by pull-down menus on the banner of the spectrum window (see more discussion of windows for spectra below).

The new map button creates a new window on the screen for display of a map or a tuned display. The map or tuned display is linked to the desired windows in the energy and x-y spectra from a station by the pull down menus on the banner of the map (see more discussion of maps below).

At the end of the row of function buttons is a status bar that shows the current mode of MpSys. In this example the scan is stopped and the last run data file has been closed. When data collection is in progress this changes to show the status.

When a run is in progress, this is what the task bar looks like. In this case the data is going into file videop2.evt

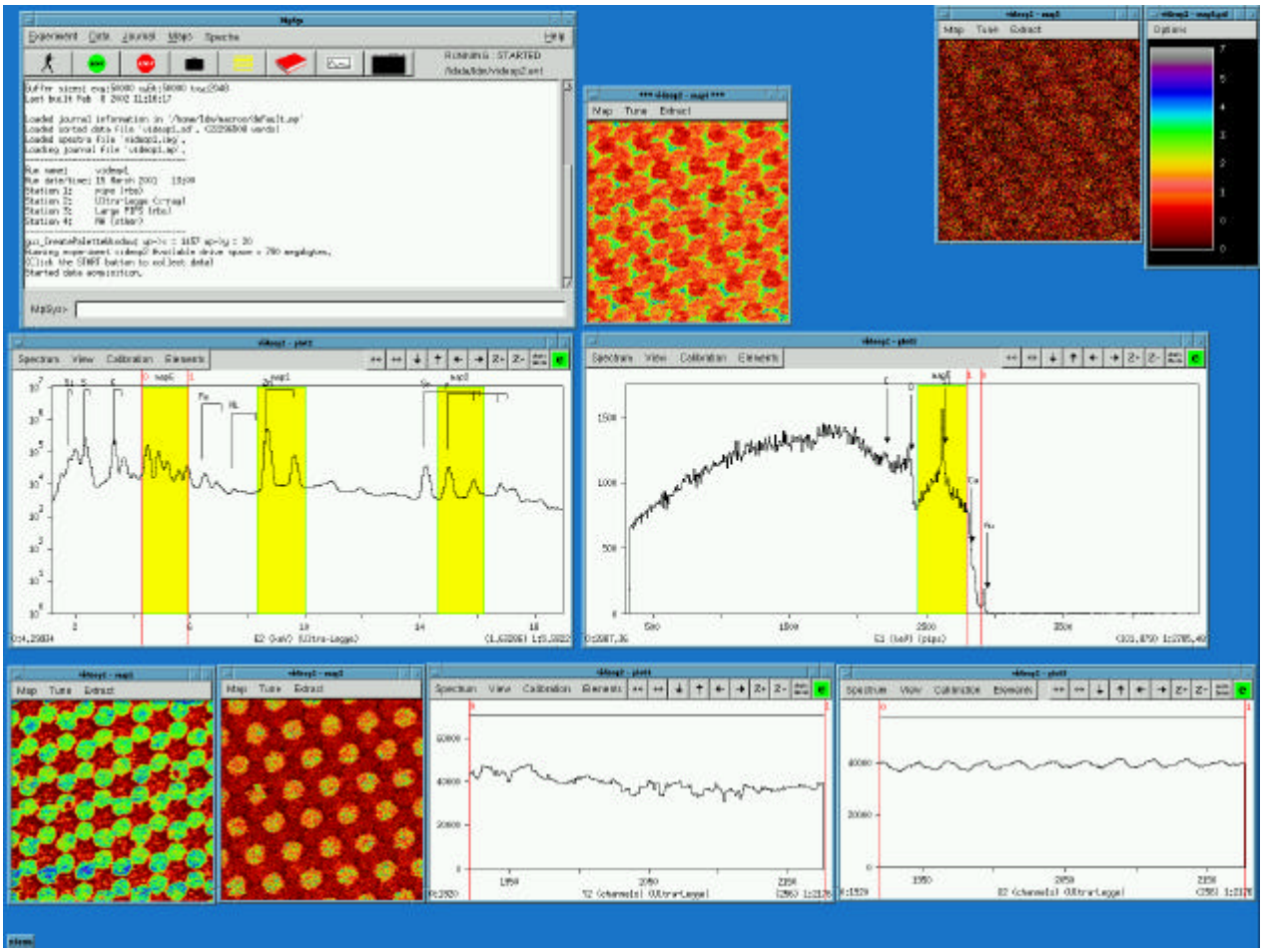


During a run, many map and spectrum windows will be on the screen. The configuration of these windows is saved in the journal file so that the configuration may be retrieved for future analysis.

An example is shown on the next page.

Example desk top

This shows an example desk top configured by the user and stored in the journal file for future reference.



It shows a typical configuration of MpSys for data collection on two stations showing spectrum and map windows.

Starting a run

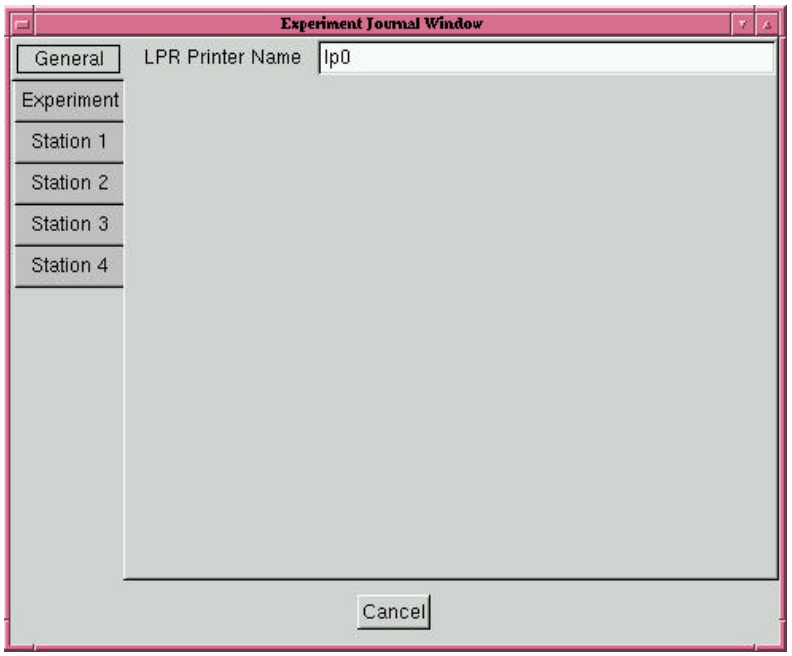
The simplest way to start a run:

1. Login to the data acquisition work station then set the directory to the local data area (creating a new subdirectory for the current project if required).
2. Start MpSys from the command prompt. If you have run before, or if there is a system default journal file, a series of spectrum windows will appear on the screen and the starting values of many parameters in the journal file will be loaded at this time. If you have not run before, create as many spectra and map windows as you like from the spectrum and map buttons. Use the pull-down menus on the task bar for the spectrum windows to connect them to the energy windows of the stations connected to detectors.
3. Click on the journal button to set the present experimental parameters such as the beam energy, beam particle and parameters for the various detectors in use. These parameters will be stored with each run.
4. Click on the run button to prepare the system for data collection. At this time the general data window from the journal file also appears to allow you to confirm or change any of the run parameters.
5. Click on the start button to start data acquisition and logging of the data to disk. Once the system is running, it is not possible to change the scan parameters. If it is necessary to change the scan parameters it is best to start a new run. In some circumstances (rare) if may be necessary to change the scan parameters during a run which can be done by stopping the run, changing the parameters, then starting the run once again.
6. At the conclusion of the run, stop the data acquisition by clicking the stop button , then clicking the close button to close off data acquisition into the data file.
7. Further runs may be done by returning to step 4.

Setting the experimental parameters in the journal file

Prior to collecting any data, the experimental parameters need to be loaded into the journal file. This is necessary because MpSys uses some of these parameters to display elemental information, calculate energy calibrations, or simply because it is good practice to keep a detailed record of the experimental parameters with each data file for future reference.

The journal file is accessed from the journal file button on the main window. This causes this pop-up window to appear.



In the left hand panel are the categories within the journal file that hold the experimental and other parameters associated with the run. They appear like the "directories" in a file manager and may be expanded by clicking on the category headings in the list.

The expanded list of parameters associated with each category appears in the right panel.

Experiment

Clicking on the **Experiment** category pops up the same window:

Under **Experiment** appear the parameters associated with the individual run. In the example shown above you can see the parameters in the General category. These are:

Run Name The name of the run and contains the file stem used to generate the file names for the event-by-event data file, the spectra file and the journal file. In the example shown these would be videop1.evt, videop1.img and videop1.mp. If you select a **Run Name** with an embedded number, MpSys will automatically increment this number for subsequent runs. So in this example, MpSys will choose video2p as the next **Run Name**.

Run ID A global run identification number that is stored in a system area and incremented each time any user starts a new run on the workstation. This allows each run to be uniquely identified if the user reuses old run names.

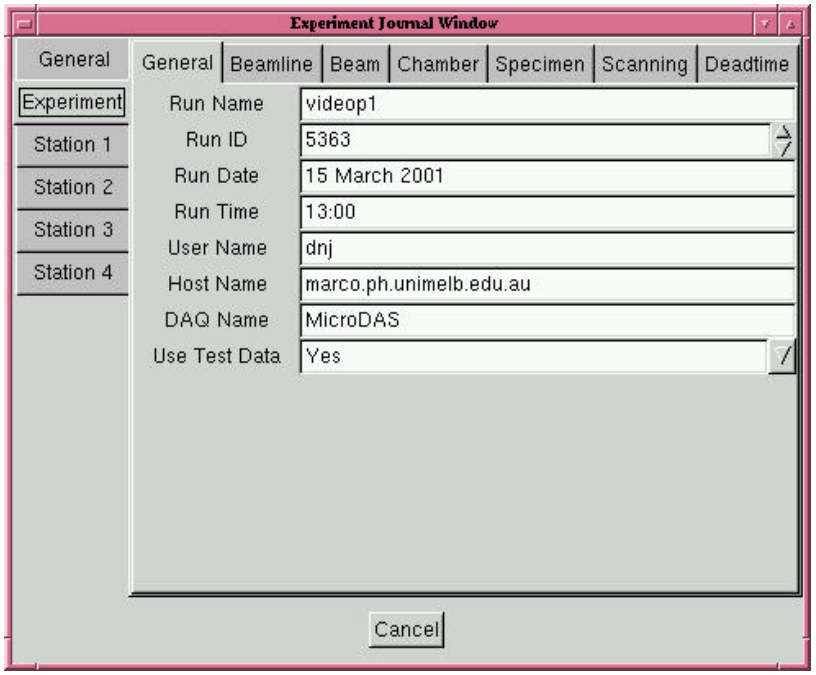
Run Date The date of the run.

Run Time The time of the run.

User Name The login name of the user.

Host Name The name of the workstation used to collect the data.

DaQ Name The name of the data acquisition system used to collect the data. This is to distinguish the system from earlier versions of the hardware used with MpSys.



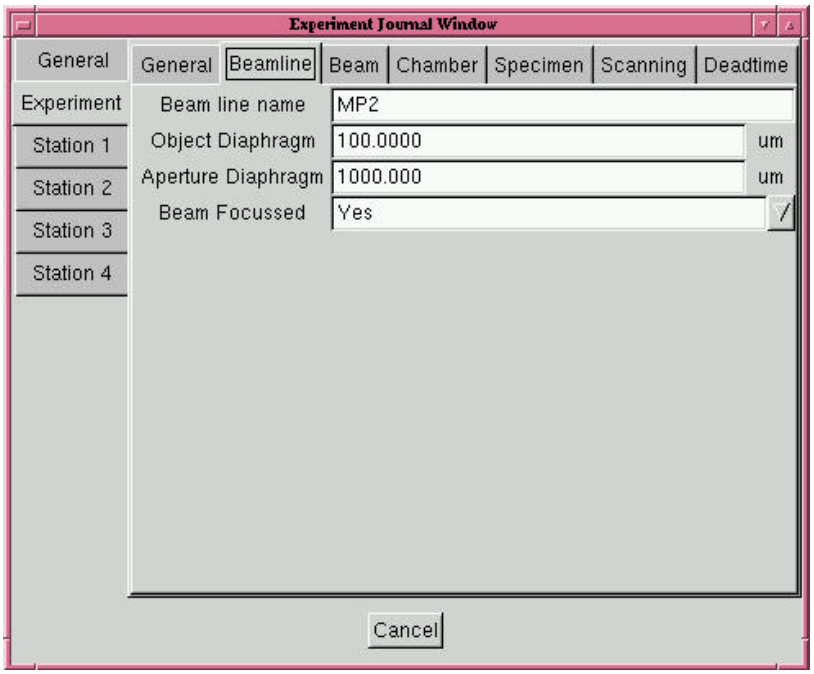
The **use test data** window is used to connect MpSys to a file in the current directory called testdata.evt for off-line collection of data. This is useful for demonstration or diagnostic purposes.

This reveals the sub-categories of the **Experiment** category. For use of some of the built-in features of MpSys for elemental identification in spectra, it is essential that the information in these sub-categories be set correctly.

The parameters associated with the elemental identification features are enclosed in a box in the discussion to follow.

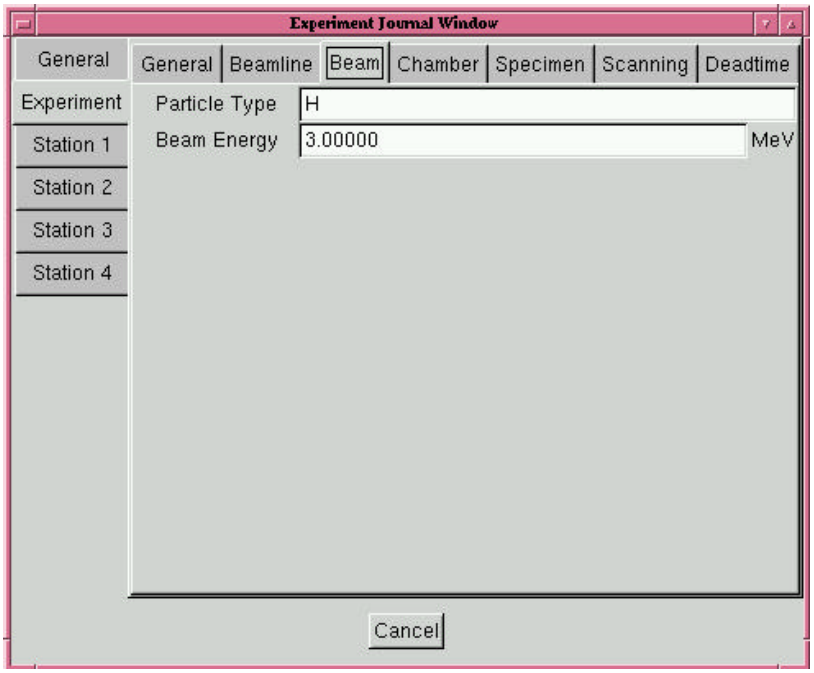
Beam Line

The first is **Beam Line** which holds parameters associated with the beam line itself.



Beam

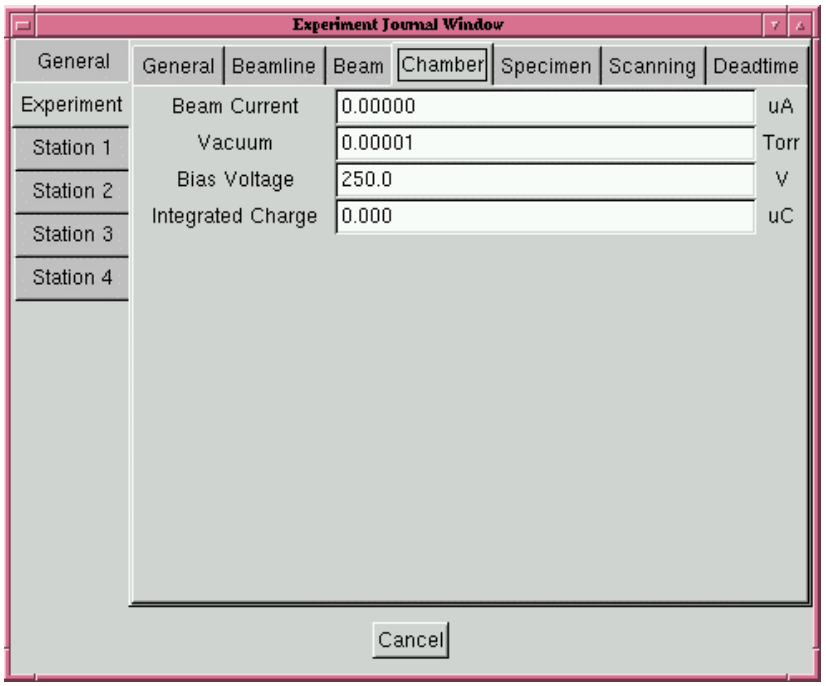
The next is the **Beam** category:



Both the beam particle and the beam energy are required for the elemental identification features of MpSys.

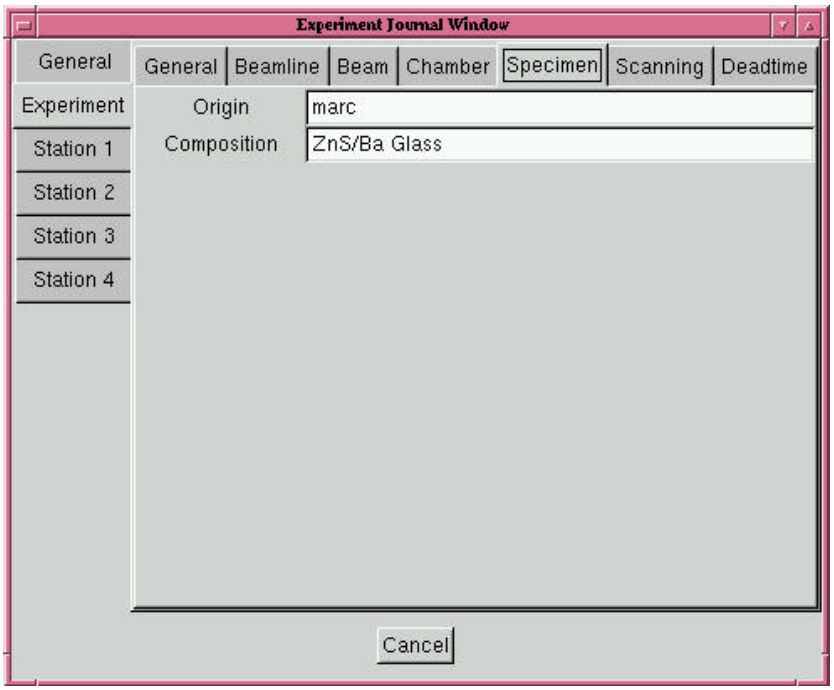
Chamber

The **Chamber** category provides slots to enter parameters associated with the charge integration system.



Specimen

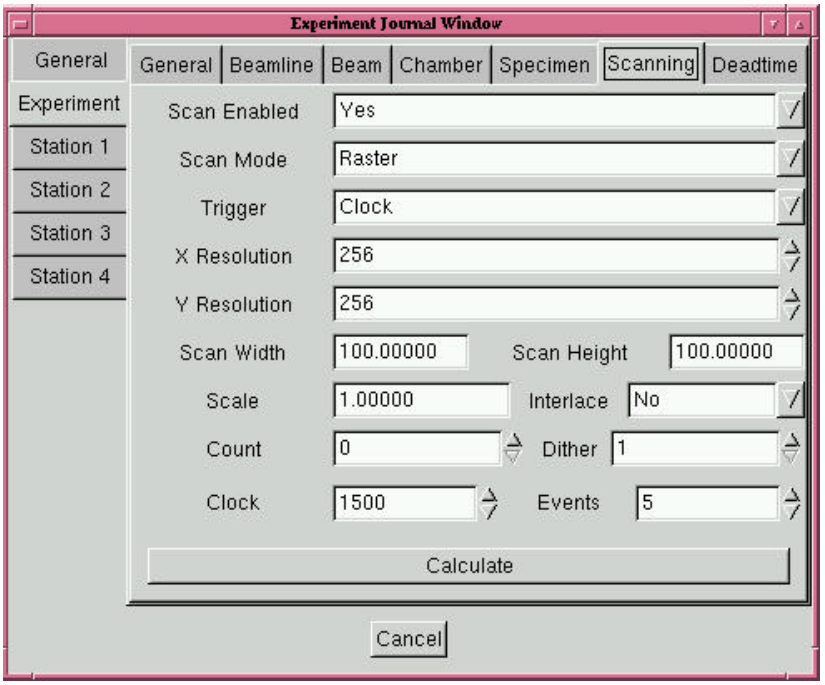
Under the **Specimen** category there are slots for putting information about the curation of the sample.



Scanning

The **Scanning** category is very important because it sets the scan parameters which are sent to the workstation I/O card which then provides the scan signal to the scan amplifier.

This screen can also be accessed through the scan button on the main menu task bar.



The slots listed on the right panel have the following properties:

Scan enabled Allows switching off and on the scan in software.

Scan Mode MpSys provides two different scan modes.

Raster is the most commonly used mode. In this case the beam commences its scan in the top left corner of the scan area, then continues in the horizontal direction to the right, dropping to the next scan line when the right edge of the scan region is reached and returning to the left edge. There is no flyback.

Triangle is an alternative mode where the scan covers a "lissajous" like figure, allowing a faster sampling of the scan area compared to the **raster** mode.

X-resolution Sets the number of pixels in the horizontal direction. Note that this does not change the scan size, it sets the resolution of the scan.

Y-resolution Sets the number of pixels in the vertical direction. Note that this does not change the scan size, it sets the resolution of the scan.

Examples: If the scan size is set to 256 µm, then a X-resolution of 256 will set the step size between pixels to 1 µm. A X-resolution of 128 will set the step size between pixels to 2 µm.

Width The width parameter sets the width of the scan area. Values of the width parameter less than 100 will proportionately narrow the scan width.

Height The height parameter sets the height of the scan area. Values of the height parameter less than 100 will proportionately compress the scan height.

Scale The scale parameter allows fine control of the entire scan size.

Count The count parameter is uses when it is desired to produce just a precise integral number of passes over the scan region. If count is zero, the scan continues rastering indefinitely.

Example: Setting the count to 2 will cause the beam to raster to the bottom of the scan area, then raster back to the top whereupon it will stop.

Dither The dither parameter causes the rastering to skip an integral number of rasters on the first pass, on the return pass the missing rasters are gradually filled in. This option is useful is it is desired to pass quickly over a region to check if the scan region is on the correct place.

Interlace Setting **interlace** to **no** causes the scan to raster uniformly from top to bottom and return. With interlace set to yes, the scan does every second raster on the way down, on the return the missed rasters are filled in.

Trigger Advance of the scan from pixel to pixel is triggered by several methods.

Clock advances the scan after an elapsed time. A slot for entering the desired dwell time is provided. It is best to use a dwell time greater than 1000 ms if magnetic scanning is used. This will prevent eddy currents in the beam tube from causing double imaging of edges in the scan region.

Charge advances the scan after a set number of charge counts have been counted. A slot for entering the desired number of charge counts is provided. An appropriate charge digitiser unit is required to be connected to the charge input on the rear of the MicroDas interface unit. A suitable unit would have a sensitivity of at least 10^{-12} Coulombs/pulse. **Charge** mode is the most common mode of operation of the system.

External advances the scan after a set number of external events have been counted from the external input of the MicroDas unit. In other respects the operation of the **External** mode is similar to the **Charge** mode.

Events advances the scan after a set number of energy events have been counted. This mode is typically used with IBIC experiments where the sample can be expected to produce an IBIC signal at every pixel of the scan. In an IBIC experiment the contrast in the image is provided by variations in the energy of the signal, not in the intensity of the signal. This mode is also appropriate for STIM/CSTIM experiments. Note that only a single station should be in use for this mode to work correctly.

The true scan size of the beam on the sample will be given by the following formula:

$$\text{X size (micron)} = k.N_x.(\text{MicroDas X gain}).(\textbf{Width/100}).(\textbf{Scale})$$

$$\text{Y size (micron)} = k.N_y.(\text{MicroDas Y gain}).(\textbf{Height/100}).(\textbf{Scale})$$

Where

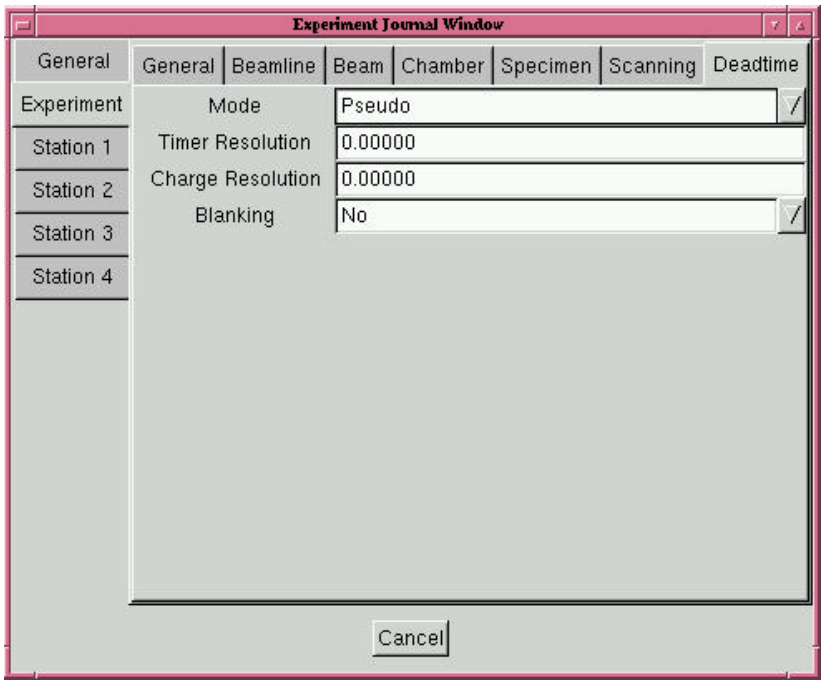
k	is a constant that depends on the beam energy, particle and lens system magnification (k=1.98 for 3 MeV H^+ ions)
N_x	is the number of turns on the x-scan coils
N_y	is the number of turns on the y-scan coils
MicroDas X gain	is the setting of the analogue gain potentiometers on the front of the MicroDas unit for the x-direction
MicroDas Y gain	similarly for the y-direction
Width	is the parameter from the scan category window above

Height similarly

Scale similarly

Deadtime

The **deadtime** category window requires a full separate manual to discuss the meanings of the deadtime options.



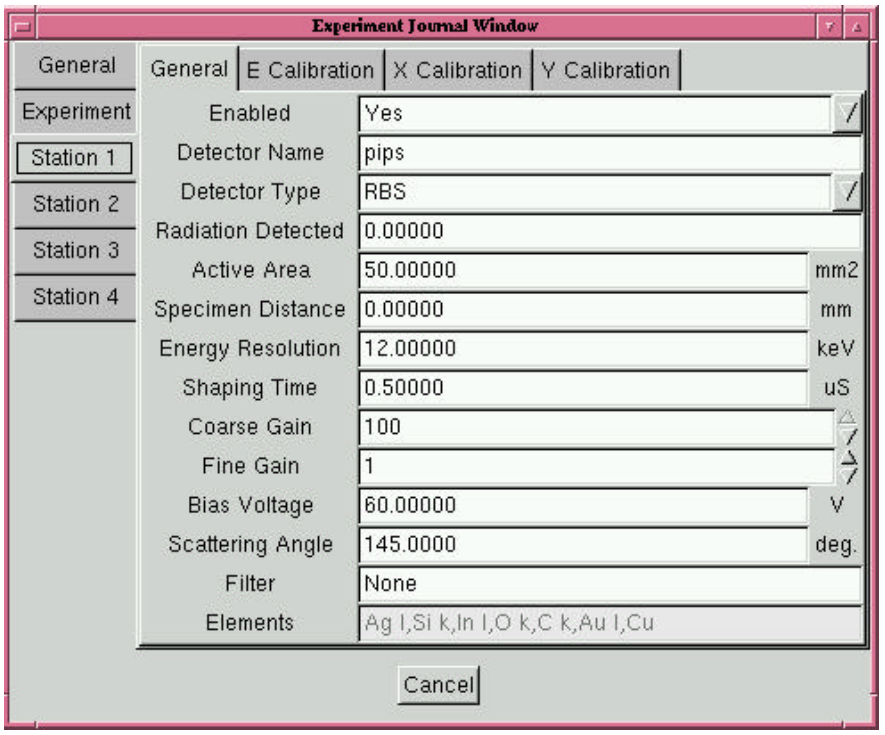
Under **Mode**, selection of **pseudo** causes the beam to dwell for additional time at each pixel to compensate for the deadtime of the ADC units and the computer. In this way, data collected when only a single station is active will be fully deadtime corrected. This is the most common mode of operation.

For the other deadtime modes see the separate manual.

Station – General

MpSys collects data from up to four stations. The parameters of the detectors connected to the stations must be entered on the screens under the **General** category for each station. The detector parameters are required for the MpSys element identification features to work correctly.

This screen shows the **General** category for station 1. The screen is the same for the other three stations.



The parameters for the general category for the detector connected to station have the following attributes:

Enabled This slot allows the station to be disabled in software (feature not implemented in MpSys version 3.1)

Detector Name The name of the detector can be entered in this slot. This name will be used to label the axes of the E, X and Y spectra associated with this station.

Detector Type Selects from rbs, X-ray or other. This sets the data base to be used for elemental identification features.

Radn. Detected Not used.

Active Area The area of the detector facing the sample.

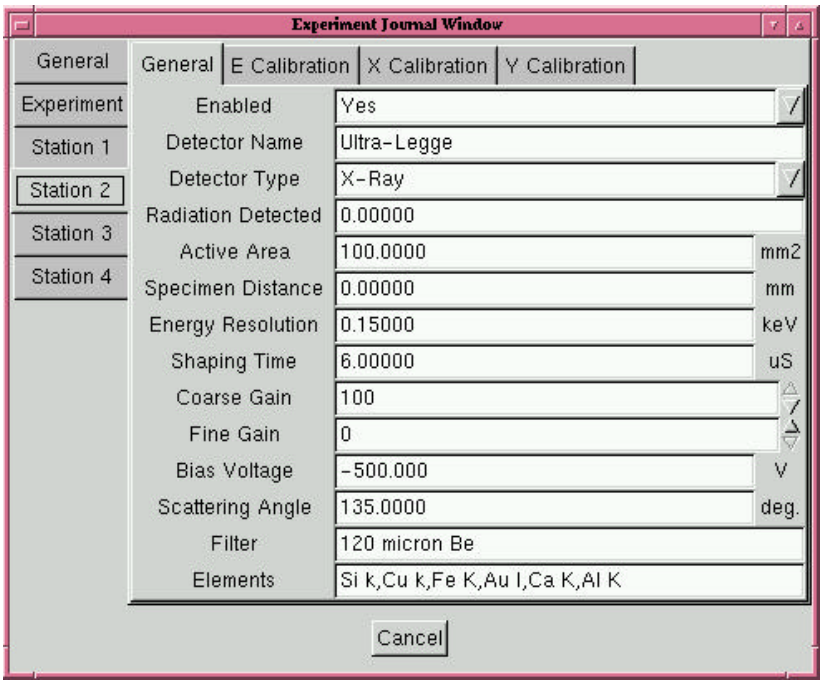
Specimen Dist The distance of the detector from the sample. Used to calculate the solid angle of the detector from its active area.

Energy Resolution The energy resolution of the detector. Used in the automatic elemental mapping feature to set the width of the window placed in the energy spectrum about the element signal.

Shaping time The shaping time used on the spectroscopy amplifier for the detector.

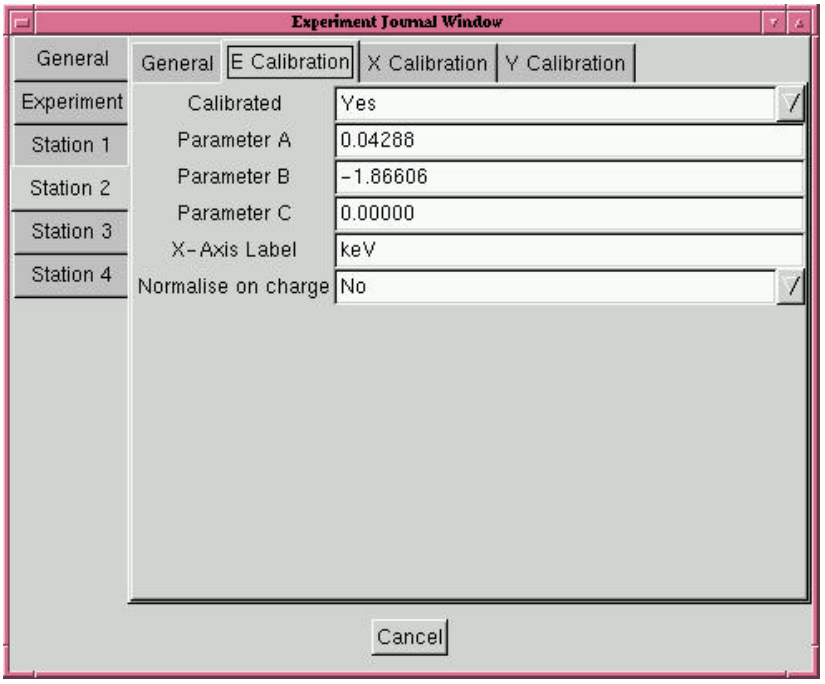
Coarse Gain	The coarse gain used by the spectroscopy amplifier for the detector.
Fine Gain	Similarly for the fine gain.
Bias Voltage	The detector bias voltage.
Scattering Angle	The angle between the straight through direction of the incident beam and the centre line of the detector in degrees. In this convention the beam approaches the sample from a scattering angle of 180°. Backscattering detectors therefore have scattering angles between 90 and 180°.
Filter	The name of any filter used over the front of the detector.
Elements	The list of elements associated with the station that will have their signals labeled on the energy spectrum. These elements are set using the element selection menu button E on the task bar associated with a spectrum window.

A second example of the general category is shown here for an X-ray detector.



Station – E Calibration

The E Calibration screen allows the energy calibration for the detector to be reviewed. The best way of determining the energy calibration for the detector is via the procedure discussed in the next section of this manual.



The energy calibration applied to the data is shown in this window:

E (keV) = Channel number***a** (keV/ch) + **b** (keV) [+ **c***(Channel number)^2]

Station – X Calibration

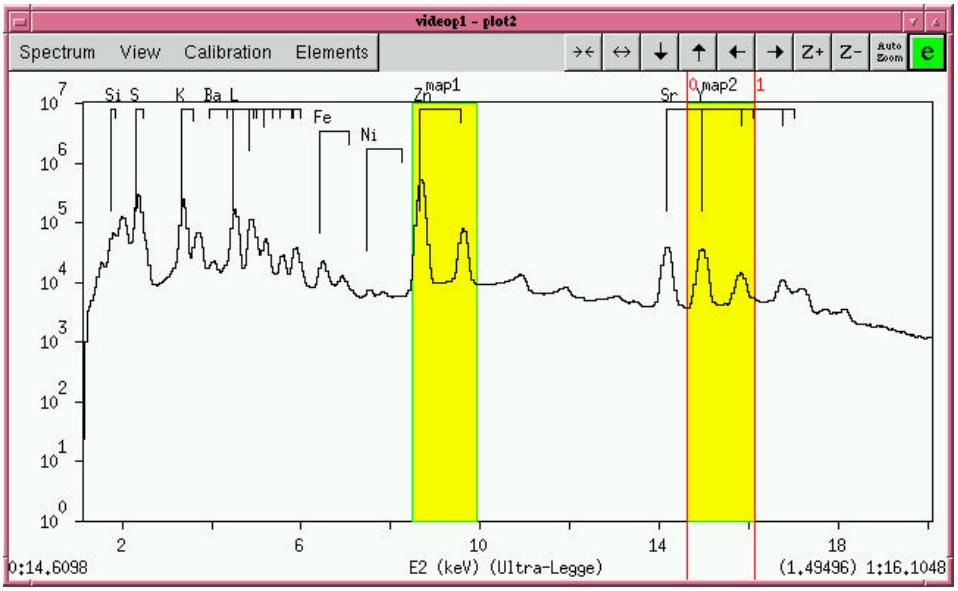
The calibration screen for the X and Y axes are very similar to that for the energy spectrum.

Working with a spectrum in a spectrum window

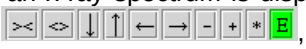
When working with spectra, it is necessary to ensure the spectrum you wish to work with has the "focus". This will ensure all commands and modifications apply to the correct spectrum. The spectrum with the current focus is shown by the row of astericks (***) flanking the name of the spectrum in the title bar.

To give the spectrum the focus, simply left click the mouse on the spectrum area (not the frame or title bar).

Spectrum control buttons



To display a spectrum on the screen, first create a new spectrum window with the new spectrum button . Then the spectrum associated with a particular station may be displayed in the window from the **spectrum** button on the spectrum window task bar. From the pull down menu, select the station number of interest, then the E, x or y spectrum to be displayed.

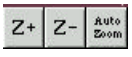
In the example shown here, an x-ray spectrum is displayed. The spectrum control buttons on the top right of the task bar,  , have been used to expand the spectrum for a closer look a small range of energies. These spectrum control buttons have the following functions:

Contract the spectrum so more of the spectrum is visible in the window.

Expand the spectrum so that a narrower range of the spectrum is visible in the window.

Raise or lower the upper limit of the spectrum to raise or lower the spectrum height.

Move the spectrum to the left or right in the spectrum window.



Access the element selection menu to allow the list of elements associated with the current spectrum to be changed. These elements are displayed as arrows at the surface energy (for RBS spectra) or as x-ray lines (for x-ray spectra). The element selection menu allows elements to be listed, deleted or extended. The use of this feature is discussed following the next section on calibration of an energy spectrum.

Calibrating an energy spectrum

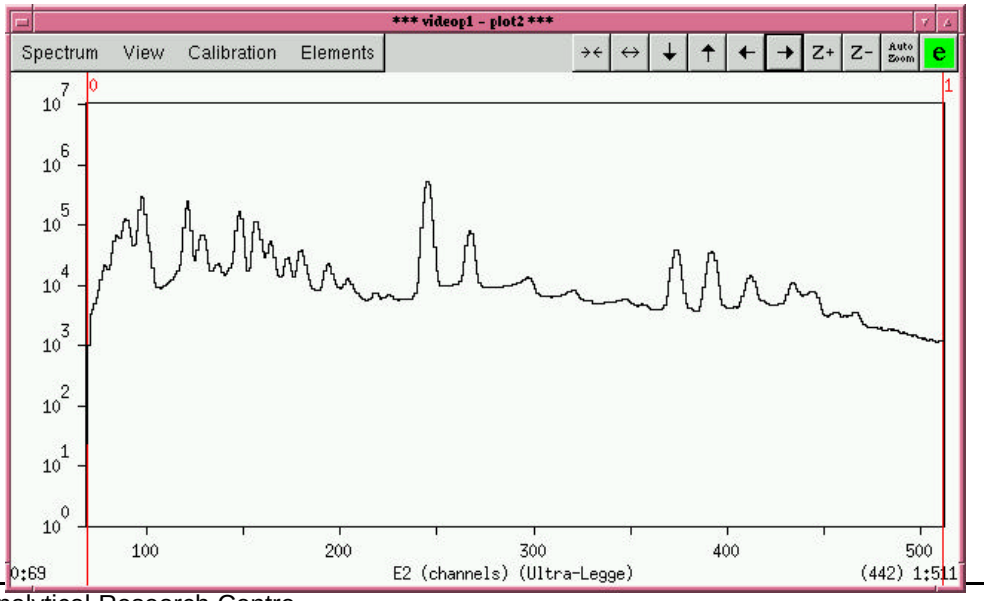
Before calibrating an energy spectrum, the experimental parameters must be set in the journal file. This may be done by clicking on the journal button and entering the appropriate parameters in the pull-down menus. The energy calibration may then be determined from the spectrum of a standard sample by identifying two known points in the spectrum (commonly peaks or edges).

MpSys has two built-in data bases. One is for RBS spectra and the other is for PIXE spectra. If the station corresponding to the spectrum you wish to calibrate is for either of these types of data, you can easily use the built-in data bases to calibrate the spectrum by just identifying two features in the spectrum that correspond to signals from two different elements. MpSys will then identify the correct energy of these two points from the appropriate built-in data base and calculate the correct energy calibration.

If your spectrum does not correspond to either of the built-in data bases, then the calibration can still be performed by identifying two features in the spectrum for which the energy is known. MpSys will then accept the correct energy for these two features and calculate the correct energy spectrum.

This process is summarised as follows:

1. Set experimental parameters using journal file button .
2. In the sample chamber, select your standard sample which should contain two or more known elements. Suitable samples are a matter of taste, but a quartz screen or a fragment of old SSB detector (silicon substrate with thin gold layer) are suitable.
3. Expose sample to the beam and collect an energy spectrum from all active stations by following the sequence new run , start , (wait), stop and close .
4. A spectrum similar to this example (PIXE from an Al/Sr sample) will appear. Notice that this is an uncalibrated spectrum because the horizontal axis is in units of channels.

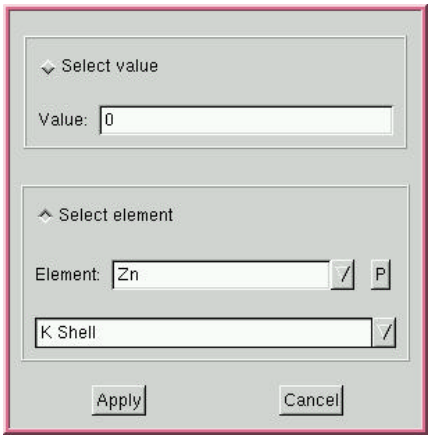


5. If necessary, clear the old energy calibration by clicking on **calibration** on the task bar of the spectrum window, and select **uncalibrate**.

6. Use the mouse to point to the first signal in the spectrum that corresponds to a known element in the standard sample. In this case the large peak on the left is known to correspond to aluminium (Al). It is not necessary to click on the mouse buttons at this stage, but this can be useful to provide a visual confirmation of the feature selected.

7. With the mouse pointing at the known signal, type **M** on the key board (either upper or lower case is fine). Be careful not to move the mouse while you do this.

8. Now MpSys asks for the element that corresponds to the **m**arked signal with a new pop up window. This window allows the correct energy associated with the **m**arked signal to be identified. There are two options selected by the top buttons:



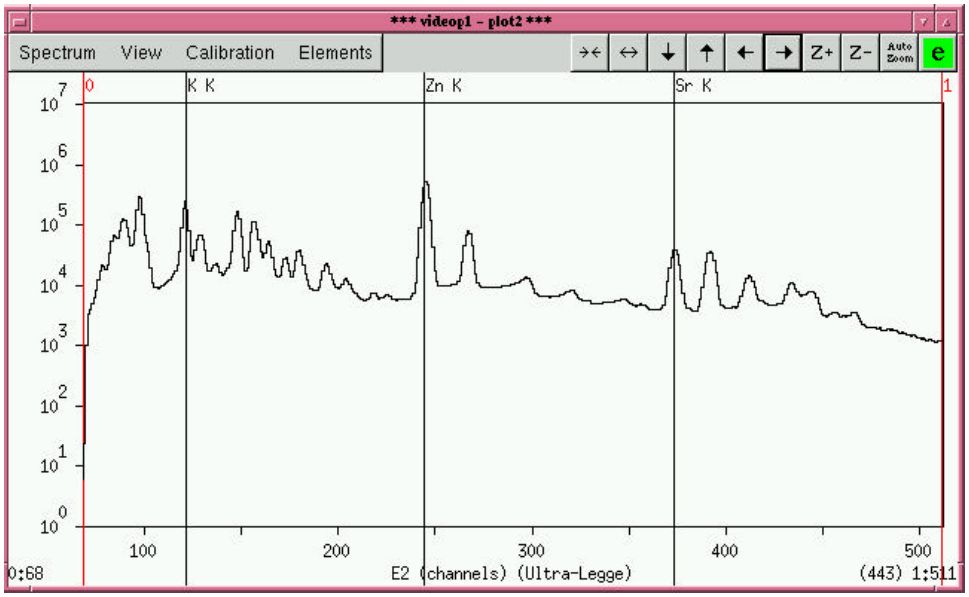
   - The left button, **Element**, allows the chemical symbol for the element to be selected.
   - The right button, **Value**, allows the correct energy of the signal to be entered by hand in the value box.

   With the **Element** option, and the station set to X-ray, it is possible to choose the chemical symbol corresponding to the **m**arked signal by typing directly into the box.

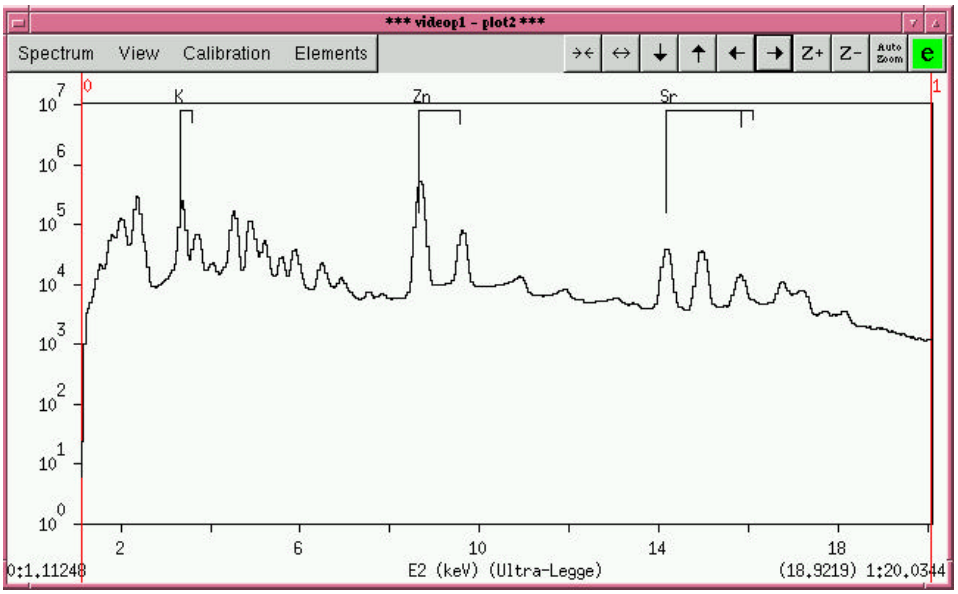
   In the example shown here, the element has been typed into the element box and the relevant x-ray line for the element has been selected from the **Shell** window. Of course, not all elements have all K, L and M lines, likewise some elements have entries in the data base for more than one x-ray line. Only the most intense x-ray line for the selected shell for the element may be used for calibration purposes. See further discussion of the data base below. Instead of typing the element chemical symbol, it is possible to click on the periodic table button, P, and select the element from the pop-up periodic table.

   The remainder of the calibration is as before.

9. With the first element marked, MpSys marks the element and is now ready to receive a second marked element. So steps 6 to 8 must be repeated for additional elements. When the additional elements are **m**arked, they appear in the uncalibrated spectrum.



10. With two or more elements marked, the calibration can now be calculated and applied to the spectrum. This is done from the **calibration** option on the task bar of the spectrum window. Select **calibrate** on the pull-down menu.
11. Now the calibration is applied to the spectrum and the caption on the horizontal axis changes as an indication that the spectrum is now calibrated. In addition, the actual values of the calibration can be seen in the relevant slots in the journal menu accessed by the journal file button .



12. Once the calibration has been applied, the new calibration must be saved in the journal file associated with the run. This is done from the pull down menu associated with journal on the main menu. If the calibration has been done during a run, then the **close** command (or button click) automatically saves the calibration in the journal file.

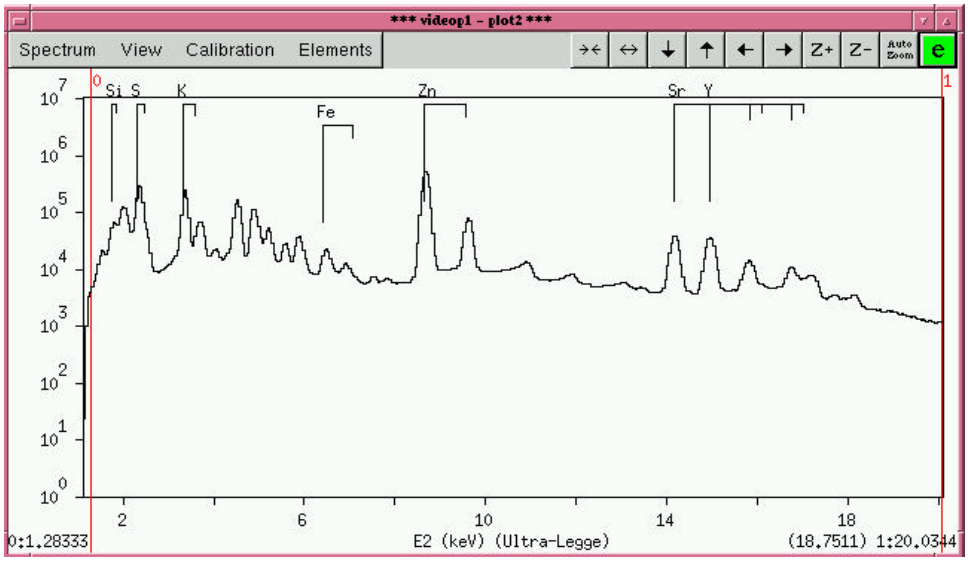
Once the spectrum is calibrated, the list of elements associated with the station will be displayed. See the discussion of spectrum windows for more information.

Element labels

For RBS and x-ray type detectors, it is possible to label the position of element signals in the corresponding spectrum. This feature only works of the correct experimental parameters have already been set in the journal file and the energy spectrum is calibrated. As discussed in the previous sections, elements are associated with a spectrum using the element selection menu.

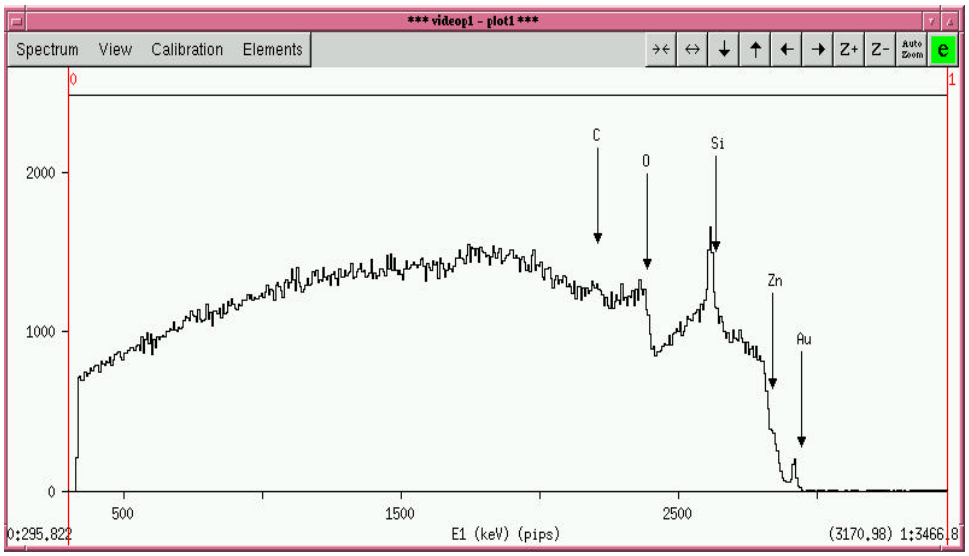
The x-ray line labels

In an x-ray spectrum, the x-ray lines for a selected element are drawn as a mini-intensity histogram. For elements with more than one x-ray line corresponding to decays to the selected shell (the "K" lines, "L" lines or "M" lines) the x-ray lines are drawn with a length proportional to the intensity of the line. The following example shows this:



RBS surface energy labels

In a RBS spectrum, the elements are identified by arrows drawn at the surface energy for each element selected.



The element selection button is discussed in the next section.

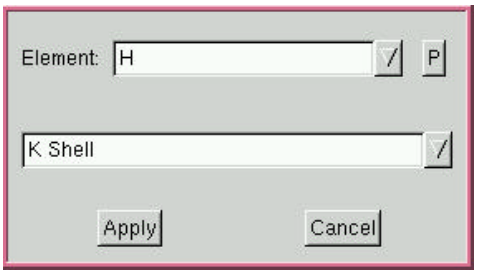
Using the element selection button

The element selection button allows the user to select the elements to have their signals labeled in the spectrum. For this feature to work correctly the essential experimental parameters must already have been set in the journal file.

Clicking on the element selection

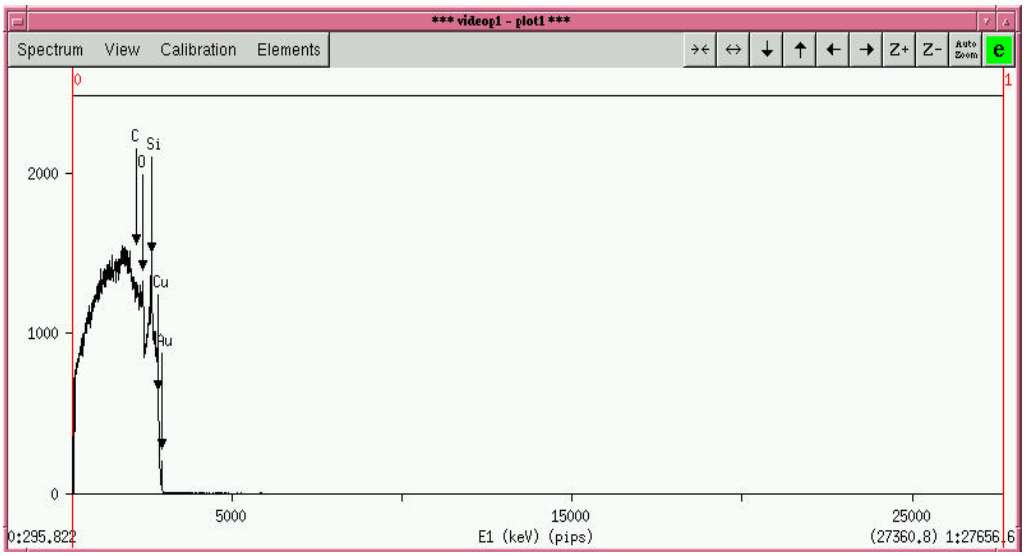
button pops up the list of elements associated with the spectrum. Clicking on **add** allows a new element to be added to the list. Clicking on **delete** deletes the highlighted element from the list.

The **add** button pops up a new window where the name of the element can be specified by either typing in the chemical element symbol or selecting the periodic table and clicking on the desired element.

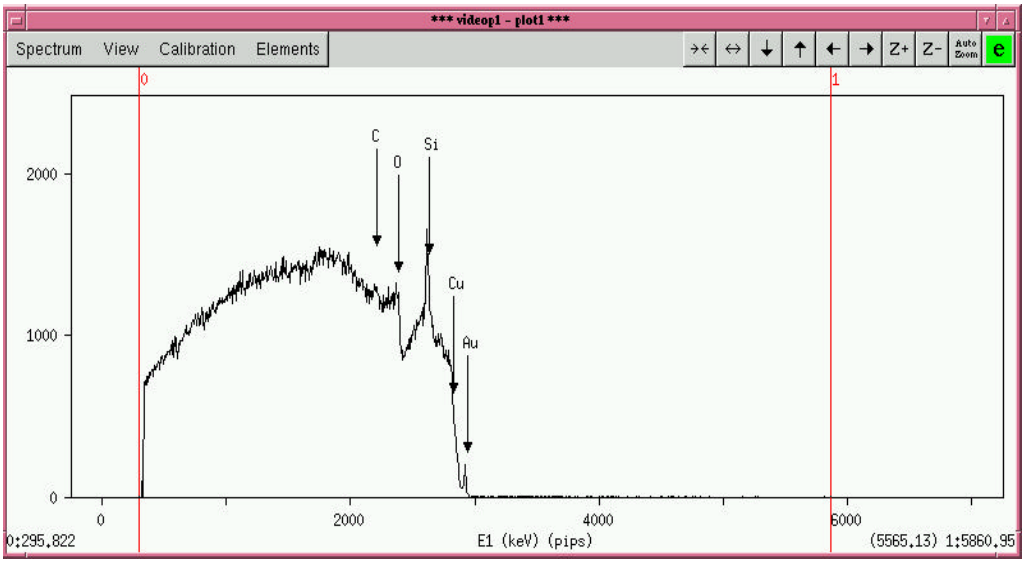


Resizing spectra

A complete 8k channel MpSys energy spectrum typically has the interesting part compressed into a relatively small number of channels. So the full spectrum needs to be expanded.



This example shows the situation where a 1 k channel RBS spectrum appears in the first eighth of a full energy spectrum. The next example shows the spectrum expanded by the F3 special function key.



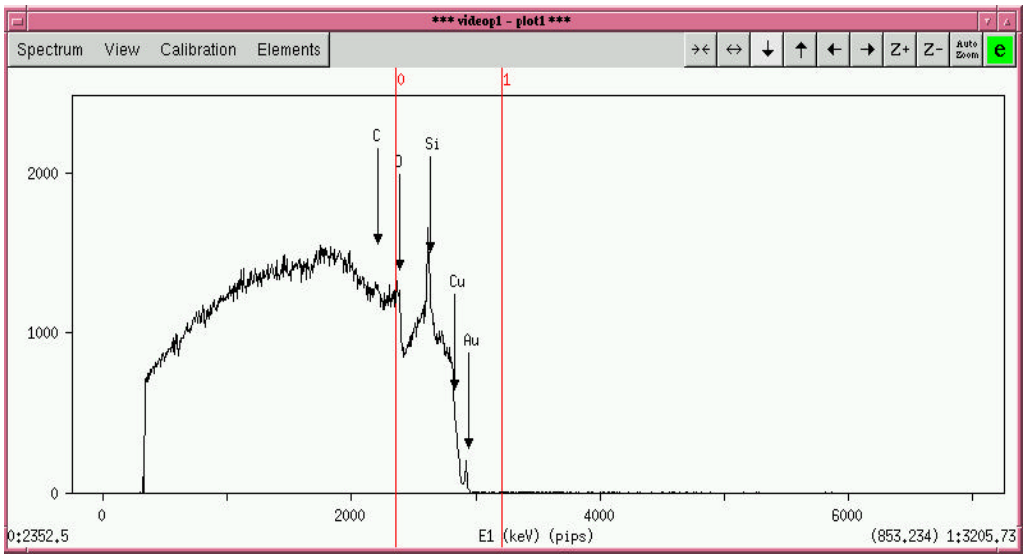
This shows that in addition to the use of the spectrum control buttons on the task bar, described in the previous section, it is also possible to resize the spectrum via mouse functions and the special function keys on the keyboard.

The special function keys are:

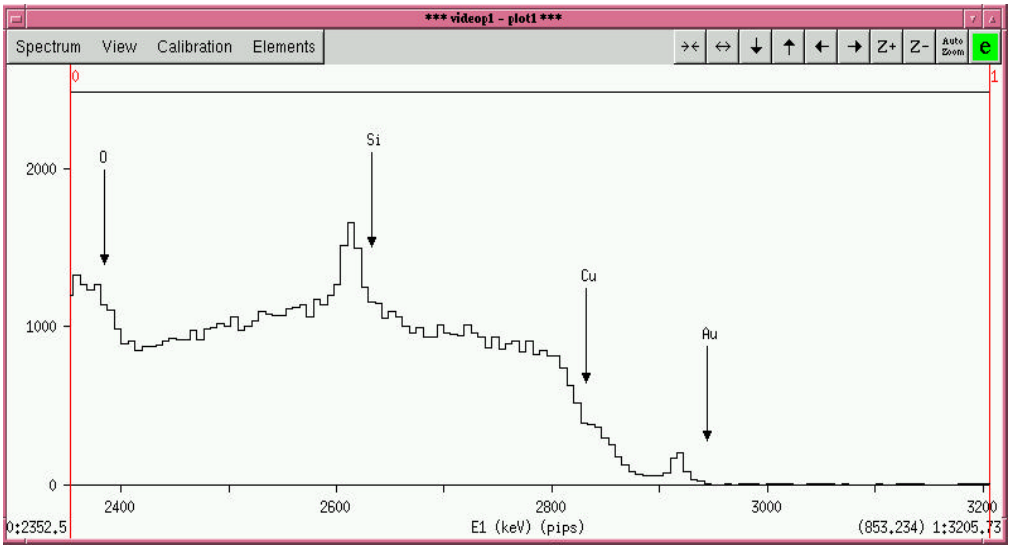
F3 Perform an automatic spectrum resize to expand the non-zero channels in the spectrum into the spectrum window. In the above RBS spectrum, the spectrum has

been resized automatically using the F3 key. The left and right edges have been set just beyond the first and last channel with non-zero counts in the spectrum

F4 Expand the spectrum so that the region of the spectrum between markers **X0** and **X1** is expanded to fill the spectrum window. Markers **X0** and **X1** are set with the left mouse button and the middle mouse button respectively.



In the example of this RBS spectrum, the markers have been positioned about the interesting region of the spectrum. Then F4 is pressed to transform the spectrum window to:



In this RBS spectrum numerous elements of interest have been identified as selected via the element selection button E.

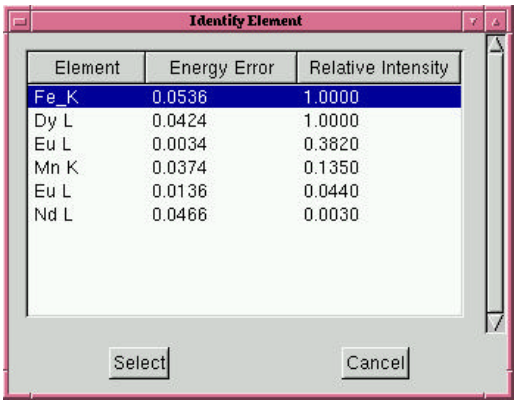
Unknown element identification

An additional way of adding elements to the element list associated with a spectrum window is to use the identify function to identify the signal from an unknown element. To use this function, the spectrum must be calibrated. See the section on spectrum calibration for more details.

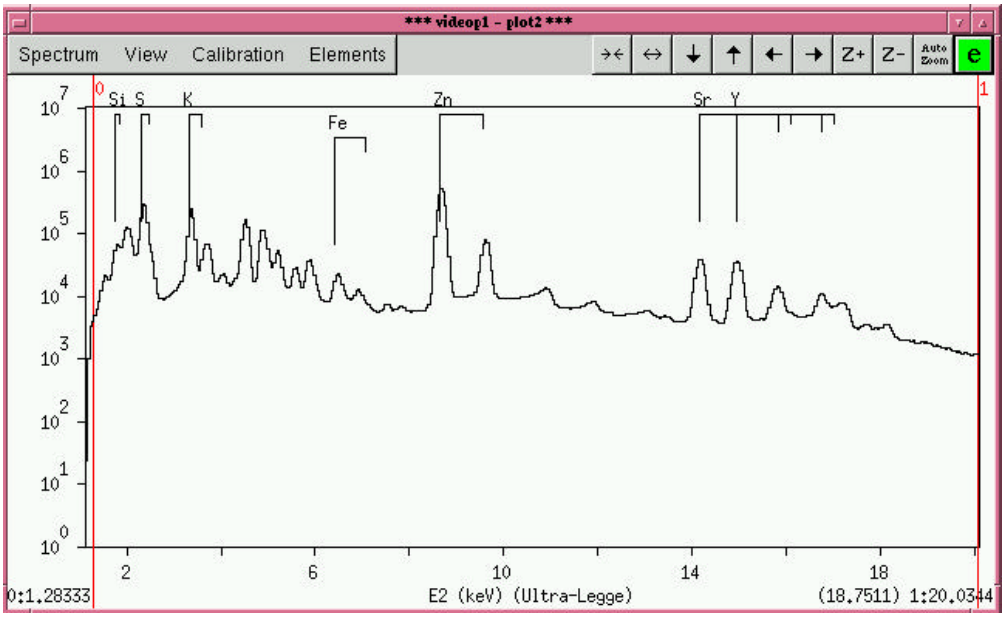
To identify an unknown element, use the mouse to point to the signal from the unknown element. In the case of an x-ray spectrum this is usually an x-ray peak. The most intense peak from the element need not be selected. In the case of RBS spectra, the signal should be the surface energy edge of the unknown element.With the mouse pointing to the unknown signal, type **I** (for **I**dentify) on the keyboard. The case does not matter.

MpSys will then search the appropriate data base for the unknown element with the energy closest to the **i**dentified signal. The energy separation between the **i**dentified signal and the data base entry will be shown in a list of possible elements. In the case of the x-ray data base, the relative intensity of each line will also be listed.

The number of possible elements in the list depends on the energy resolution of the relevent detector for the station corresponding to the energy spectrum. The energy resolution is entered as part of the journal file discussed elsewhere. A detector with a poor energy resolution will produce a longer list of possible elements. For example, typing **I** with the mouse positioned on the Fe K-alpha x-ray line in the energy spectrum below on this page produces the pop-up window shown opposite



Clearly, the most intense x-ray lines in this list have the best chance of being the unknown line. In this example, the most likely unknown line is indeed Fe, so this has been slected with a mouse click. The line is highlighted. Clicking on Ok will then cause the Fe x-ray lines to be plotted on the spectrum (see below).



Generating a tuned display or map from windows set in a spectrum

Once a run has started and MpSys has begun collecting event-by-event data to disk, it is possible to create a "tuned display" which is an intensity map of the sample in the scan region obtained from a window (or gate) placed in the energy spectrum of a particular detector.

The window may be specified by two methods.

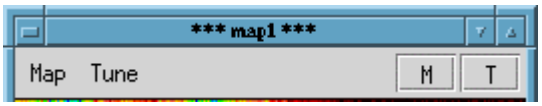
1. The first is simply to place markers in the energy spectrum around the signal you wish to map. This is done by pointing with the mouse to the left (low energy) side of the signal and clicking the left mouse button to position marker **X0**. Then point to the right (high energy) side of the signal and click the middle mouse button to position marker **X1**.

   This method can be used to set the window in any type of spectrum.

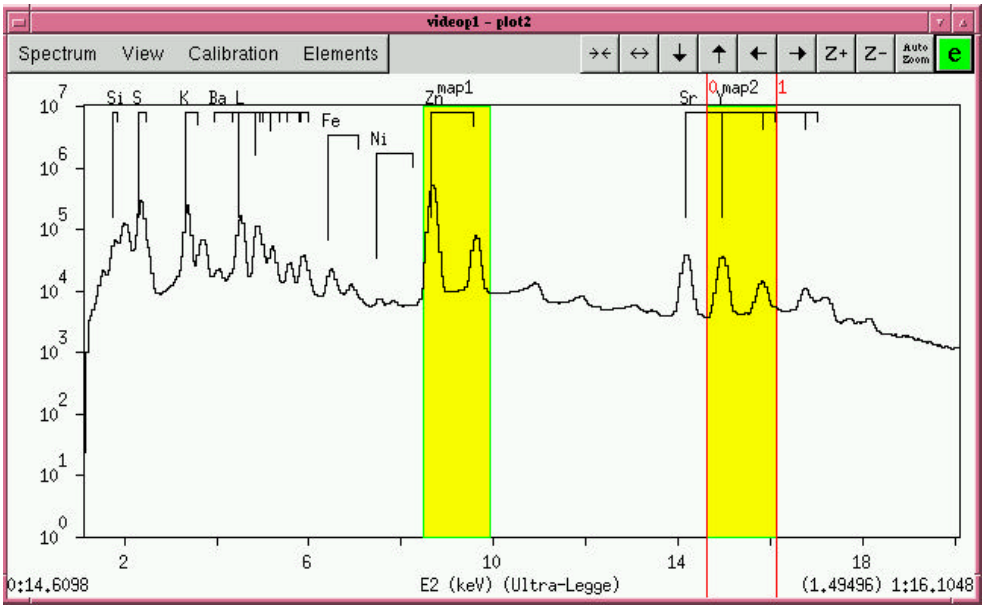
2. If the necessary experimental parameters have been entered into the journal file, and the spectrum corresponds to either an X-ray or RBS spectrum, it is also possible to specify the window by element. In this case the window is automatically positioned about the energy position corresponding to the element energy obtained from the correct data base, with a width specified by the energy resolution of the detector specified in the journal file.

To create the map, the procedure is as follows:

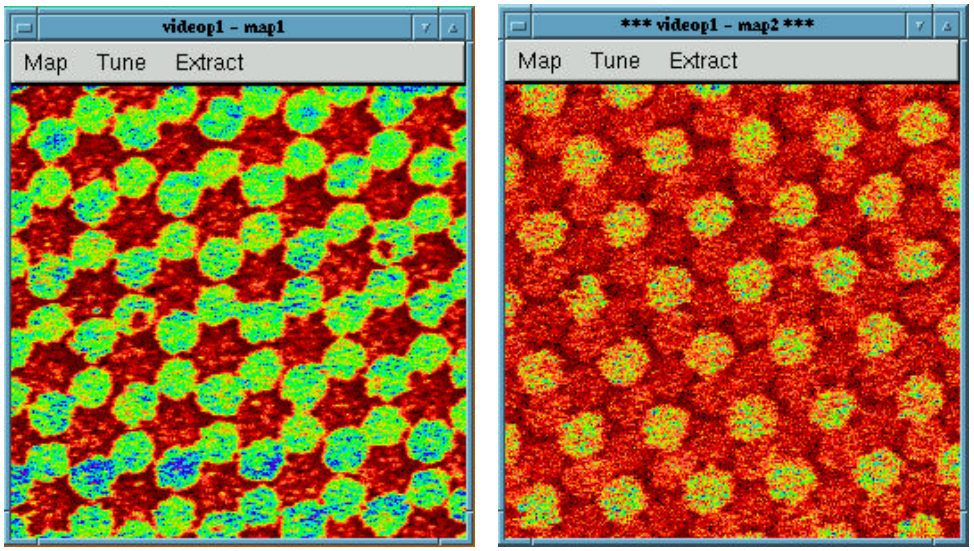
1. Position markers **X0** and **X1** in the spectrum window
2. Create a new map window with the map button, , or reuse an existing map window by clicking on it.
3. On the task bar of the map window click on the type of map appropriate for the present state of the system:
   - **Map** if you are in the process of analysing data off-line
   - **Tune** if you are collecting data and wish to see the map appear as the data is collected.



4. From the resulting pop-up window, click on the **station** number for the selected detector.
5. Choose from **markers** or **element** to select the desired window method.
6. If element was selected, a new pop up window allows the element to be selected.
7. The map specification process is now complete, a yellow bar will appear in the energy spectrum labeled with the name of the map.



In this example, element windows have been selected for Si, S, K, Ba L, Fe, Ni, Zn K, Sr and Y K. Marker windows were specified for Zn K and Y K. Two of the resulting maps are shown here.



Other map features:

The menu on the map task bar gives access to a number of different possible map operations as well as a convenient way of regenerating the map from new spectrum windows.

This table summarises the sub-menus of the **map** menu:

Menu Item	Description

Station X (Markers)	Create map using current spectra marker positions for station X
Station X (Elements)	Create map using a specified element from a calibrated spectra
Modify	Modify map parameter
Load	Create a map from a map file
Save	Save a map to a map file
Rename	Rename a map
Info	Display map parameters
Clear	Clear a map
Close	Close a map window

If the map is a tuned display which has been generated on-line, then the **tune** menu applies:

Menu Item	Description
Station X (Markers	Create a tuned map using current spectra marker positions for station X
Station X (Elements	Create a tuned map using a specified element from a calibrated spectra

Extracting spectra from Maps

One of the most powerful features of MpSys is the ability to extract spectra from regions of interest within the scan. This allows detailed analysis of features within the scan region. The fact that the entire run is stored in the event-by-event file means that the region of interest need not be specified in advance and new regions of interest can be applied to the scan region an any time after the run has concluded.

To extract a spectrum from a region of interest within the scan area follow these steps:

1. Create a map which shows the region of interest.
2. Use the mouse to start drawing an outline of the region of interest, called a **shape**, on the map. This is done by pointing the perimeter of the region of interest and clicking the left mouse button (two clicks will be necessary if the map window does not have the current focus). A line now follows the mouse pointer anchored on this starting point.
3. Continue moving the mouse around the region to point to the perimeter of the region of interest while clicking with the left mouse button. This will draw a continuous line around the perimeter of the region of interes.
4. Complete the line around the perimeter of the region of interest to conclude drawing the **shape** by clicking the middle mouse button.
5. If you make a mistake, type the command **shape –x** in the MpSys command window. This will clear the **shape** and allow you to start again.

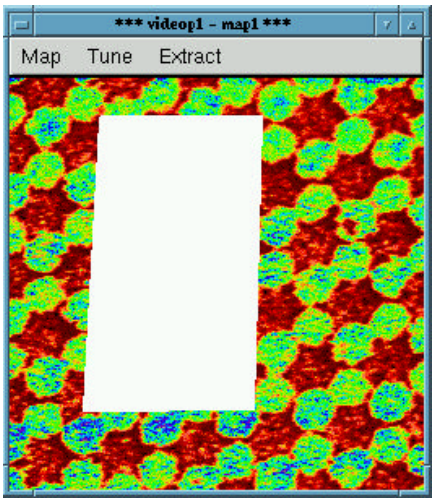
6. Extract the spectrum from the shape with the **extract** command typed in the MpSys command window. See the documentation on the **extract** command in the command section of this manual.

7. Example:

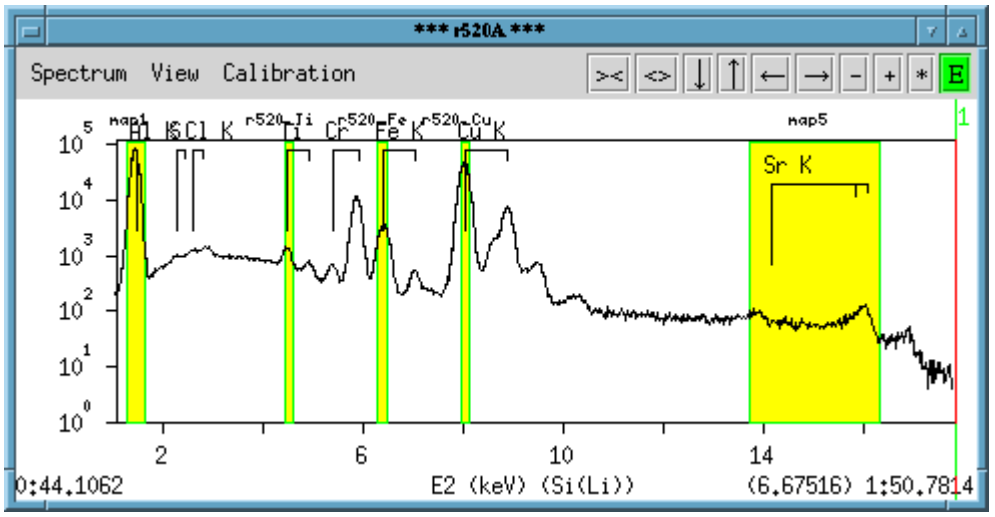
   extract –s stn=2 videop1A

   This command will extract a spectrum into a new window **videop1A** from the detector of station 2 (**stn=2**) while also shading (**–s**) the shape to confirm the region from which the spectrum has been extracted.

This shows the appearance of the map after the spectrum has been extracted:

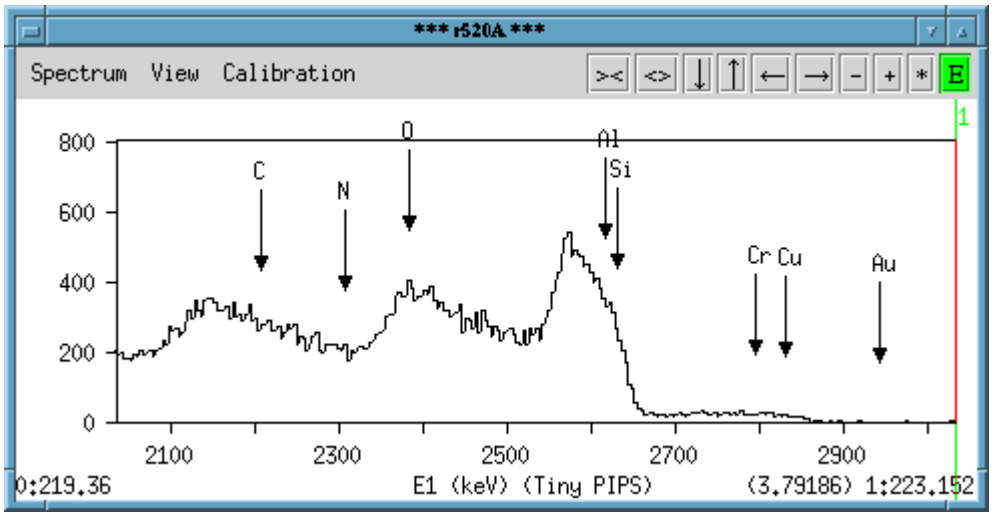


The white rectangle is the shaded **shape**. The extracted spectrum appears in a new window that pops up when the extract is complete as shown here:



Note that is not necessary to draw the **shape** in a map window derived from the same detector from which the spectrum is to be **extract**ed. So, for example, it is possible to extract a spectrum from station 1 from a map derived from a window in station 2.

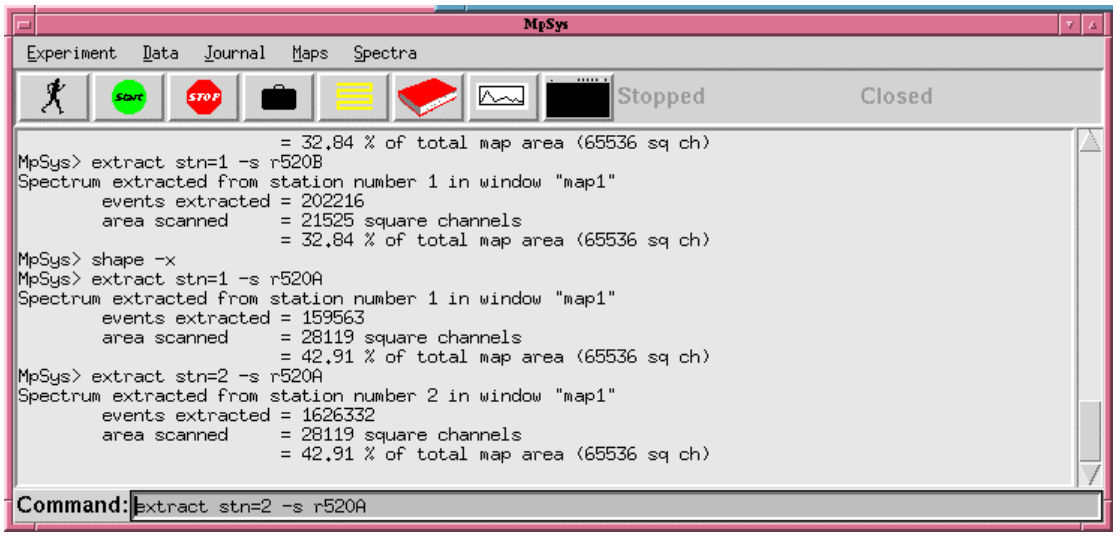
An example is shown in the next spectrum:



So this RBS spectrum applies to the same region of interest as the PIXE spectrum in the previous example.

The extracted spectrum now applies only from the sub-region of the scan defined by the shape. Therefore less than 100% of the data file has been used to make the extracted spectrum.

For future normalisation purposes, it is essential to keep a record of the percentage of the data that went into the extracted spectrum. This number appears in the MpSys dialogue screen as shown in the following example:


MpSys
Experiment Data Journal Maps Spectra
Stopped
Closed
= 32.84 % of total map area (65536 sq ch)
MpSys> extract stn=1 -s r520B
Spectrum extracted from station number 1 in window "map1"
events extracted = 202216
area scanned = 21525 square channels
= 32.84 % of total map area (65536 sq ch)
MpSys> shape -x
MpSys> extract stn=1 -s r520A
Spectrum extracted from station number 1 in window "map1"
events extracted = 159563
area scanned = 28119 square channels
= 42.91 % of total map area (65536 sq ch)
MpSys> extract stn=2 -s r520A
Spectrum extracted from station number 2 in window "map1"
events extracted = 1626332
area scanned = 28119 square channels
= 42.91 % of total map area (65536 sq ch)
Command: extract stn=2 -s r520A

The MpSys dialogue screen during the extraction of the two example spectra shown on the previous page.

The Main Menu bar

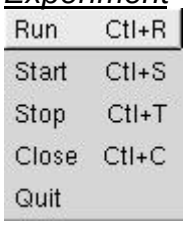
Many additional functions of MpSys can be accessed through the main menu on the main MpSys menu screen. As with all MpSys functions, these functions can also be accessed through the command window by using a MpSys command, or through a MpSys macro of MpSys commands also executed through the command window.

This section provides a brief guide to the functions which can be accessed through the main menu.

The main menu:

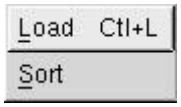
Each menu item can be clicked on to provide a pull-down list of functions.

Experiment



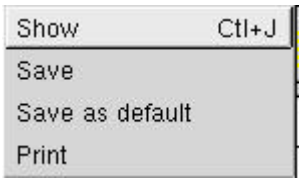
These functions control data acquisition and duplicate the functions of the menu buttons discussed earlier. **Run** to prepare a new data file to receive data, **Start** to start collecting data, **Stop** to stop collecting data, **Close** to close of the data file so that no further data can be collected into it, **Quit** to exit MpSys and return to the unix shell.

Data



The **Load** menu item allows old data to be reloaded into MpSys. The old data can be in the form of unsorted event-by-event files (.evt) or as sorted data files (.sd, .sp). The **Sort** menu item runs MpSort for sorting event by event files and producing the sorted data files.

Journal



This menu provides access to the journal file. As discussed earlier, the journal file is a macro, i.e. a list of MpSys commands, that holds the experimental parameters associated with a run.

Show provides access to the journal editing screens allowing experimental data to be loaded and reviewed. See the section titled "Setting the experimental parameters in the journal file" earlier in this manual for a detailed discussion of the journal file.

Save is used to save the journal file after any changes have been made after the run has commenced.

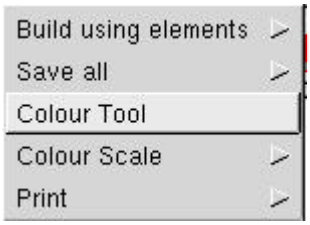
So if you make changes or corrections to the journal file, such as corrections to the energy calibration or changes to the experimental parameters, it is important to click on Save to save changes.

Save as default is used to save the current journal file as the default journal file. This will be used to configure MpSys when it is first started from the Unix shell.

Print is used to print the parameters of the journal file for future reference.

Maps

Build using elements allows all the maps associated with the present data to be recreated from the lists of elements associated with each station.



Save All allows all the current maps to be saved. The options are

Colour tool pops up the colour tool window which allows the colour mapping to be changed interactively.

Colour scale allows the pallette used to display the map files to be changed. The list of available colour scales is presented which includes grey scale and alternative colour scales.

Print allows a selected map, or array of maps, to be printed.

Spectra

Load allows a new spectrum or set of spectra to be loaded from disk.

Save saves the current spectrum or spectra in a spectrum file

Build is used to recreate the spectra associated with a particular event-by-event file or sorted data file.

4. Working with Map windows

When working with maps, as was the situation when working with spectra, it is necessary to ensure the map you wish to work with has the "focus". This will ensure all commands and modifications apply to the correct map window. The map window with the current focus is shown by the row of asterisks (***) flanking the name of the map in the title bar.

To give the map the focus, simply left click the mouse on the map area (not the frame or title bar).

Loading Maps

Maps can also be loaded from map files in existing map windows. Map files have the filename extension ".map".

To load a map file into an existing map, select the Load menu item in the Map menu of the map window. You will see a file loading dialog box appear allowing you to select which map file to load.

The map file will automatically be displayed in the map window and the window name will change to the map file name (without the extension).

Command line interface

```
load –m <filename>
load –m window="map1" <filename>
```

See section 9 for information about the format of an MpSys map file.

Saving Maps

Maps can be saved as MpSys map files, JPG files, TIF files or ASCII text files. To save a map, select the Save menu item in the Map menu of a map window and then select which file format you want to save the map as. The file formats will be one of: MAP, ASCII, JPG or TIF.

The map is automatically saved in the format specified using the map name as the filename.

If the operation is successful the message: saving map '<name>' as '<name>.<extension>' in the current directory will be displayed. Where name is the name of the map window and extension is one of: ".map", ".asc", ".jpg" or ".tif".

If the operation could not be performed then you will see the error message: save: could not save map as file '…'. <operating system reason>.

Command line interface

```
save –m <filename>
```

Printing Maps

Any map window may be printed on a postscript printer ..

```
dump type="map" windows="cu1, rb101_si"
```

The optional argument cls may also be used to specifiy the colour scale file to use when converting the map images into postscript. This is important when printing maps on a black and white printer as the printer will try to approximate its own grey scales if a colour postscipt image is supplied.

Example for printing on a grey scale printer:

```
dump type="map" windows="cu1,rb101_si" cls="grey"
```

Clearing Maps

A map window originally used to display a tuned map during a run cannot be reused to display a map generated from sorted data unless it is cleared first. Either use the command window and the command:

```
clear
```

Or click on the map option on the task bar of the map window itself.

Closing Map windows

To close a map window you can either:

1. Select the close menu item in the map menu
2. Use the kill <window name> command

Manipulating map colours

The colour scale

The colour scale used by all maps can be changed to possibly improve contrast and enhance important map features. It can be changed by loading a new colour scale into memory or by manipulating, in real-time, the colour tool.

To load a new colour scale either use the command line:

```
clscale <colour scale filename>
```

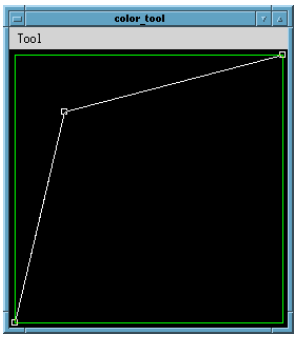
or use the **Maps** option of the task bar of the MpSys main menu. Note that the colour scale applies to all maps displayed on the screen. It is not possible to individually change the colour scale of maps.

Grey scale colour scale.

Default "spec" colour scale.

Using the colour tool

To invoke the colour tool you select the **Colour Tool** menu item in the **Maps** menu of the main window.



Command Line window on the MpSys main menu can also be used to invoke the colour tool:

```
ctool
```

The colour tool is used to interactively change the mapping between the colour scale and the intensity of the map.

Clicking on the right mouse button adds a handle to the colour line, the left mouse button can then be used to drag the line to change the mapping. Clicking on the middle mouse button clears the handle.

5. Journal file options

MpSys 3 provides a means for experimental parameter information to be recorded with actual experiment data obtained during a run. When the user loads a previous collected data set, the associated journal file will also load. A warning is given if no journal file found.

Normal use of Mpsys involves the use of a journal file.

The purpose of this section is to discuss the MpSys journal file parameters in detail. These parameters may be accessed by the users and incorporated into MpSys macros that employ the Skip programming language.

In the discussion to follow, each parameter is listed and it characteristics described. The paragraph describing each parameter has a boxed header. To the right hand side of each parameter name is the MpSys parameter variable which is used by the **set** command. For the station parameters the variables all begin with 'StX' where X is a number between 1 and 4 representing one of the four available data acquisition stations. The parameter variables can then be loaded with data with the following characterisitics:

Text any alphanumeric characters and spaces. When setting the parameter using the command line interface, it is important to enclose the value in double quotes. Example: set origin "Melbourne Australia"

Restricted text only certain words can be used for this parameter. The permissible words will be listed in the parameter description. All text entered is case insensitive.

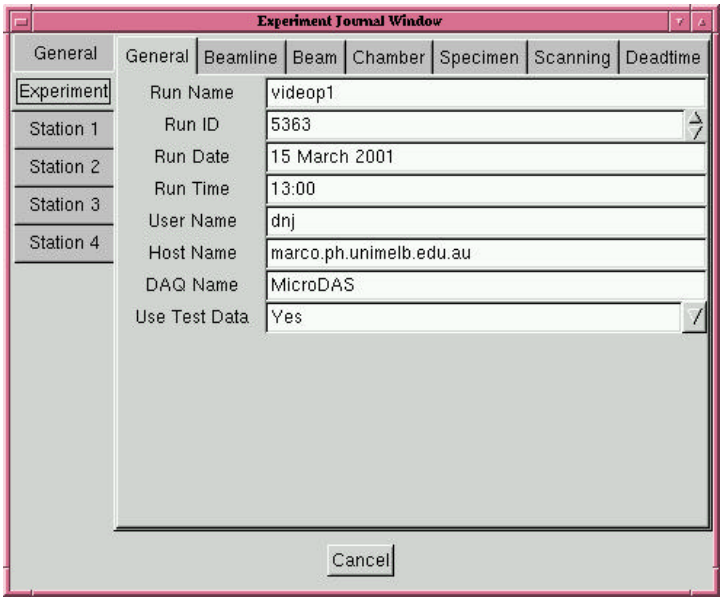
Yes/No only the text words 'yes' or 'no' (case insensitive) may be used for this parameter

Number may be integer, positive integer, real, positive real

Unit This is the standard unit for numerical field parameters. At present there is no provision for using different units for a given parameter.

Default Value This is the value assigned to the parameter when MpSys first starts. Normally it will be replaced by a setting within the user's default.mp file.

The parameters found under each of the major sub-headings of the journal file window are now described.



General Parameters

Printer Name	**printer_name**

The UNIX name of the default printer to use for plot and map dumping as well as for printing out log book reports. The value will be used as the printer name argument to the UNIX print spool program lpr. Example: if the printer_name is 'eon' then MpSys will use the command lpr –Peon when printing.

Field type: text Default value: "lp1"

Experiment Parameters

General

Run ID	**run_id**

Unique identification number obtained from the operarting system every time an experiment is run. This number is system specific so log book files used on other systems may have the same value as existing log book files on that system. If the value is 0 then it is regarded as undefined.

Field type: positive integer number Default value: 0

Run date	**run_date**

The date in which the experiment took place. When a user begins a run, the operating system will automatically put the current date in this field in the format: dd mmm yyyy. Example: 15 March 2001

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Run name **run_name**

The name of the experiment being run. This parameter cannot be changed once data acquisition has started.

Field type: text Default value: blank

Host name **host_name**

The name of the host computer that is being used for the run.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

DAQ name **daq_name**

The name of the data acquisition system that is being used for the run.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: "MicroDAS"

Beam line

Beam line name **beamline_name**

The name of the microprobe beam line used in the experiment. This will be laboratory specific.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Object diaphragm **object_diaphragm**

The diameter of the object diaphragm on the microprobe beam line.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: (m Default value: 0.0

Aperture diaphragm **aperture_diaphragm**

The diameter of the aperture diaphragm on the microprobe beam line.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: (m Default value: 0.0

Beam focussed **beam_focussed**

A boolean statement as to whether a focused ion beam was used for the experiment.

This parameter is used for information purposes only and is not used by MpSys.

Field type: yes/no Default value: yes

Beam

Particle type	**particle_type**

The type of beam particle used for the experiment. This can only be an element from the periodic table.

MpSys requires this parameter for calibrating RBS spectra.

Field type: restricted text Default value: "H"

Beam energy	**beam_energy**

The energy of the ion beam used in the experiment.

MpSys requires this parameter for calibrating RBS spectra.

Field type: positive real number Unit: MeV Default value: 0.0

Chamber

Beam current	**beam_current**

The ion beam current detected on the sample.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: (A Default value: 0.0

Scattering angle	**scattering_angle**

The angle as which the RBS detector is placed relative to the incident ion beam hitting the sample. If the value is 0.0 then it is regarded as undefined.

MpSys requires this parameter for calibrating RBS spectra.

Field type: real number Unit: degrees Default value: 0.0

Vacuum	**vacuum**

The vacuum level within the specimen chamber.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: Torr Default value: 0.0

Bias voltage	**bias_voltage**

The voltage as which the target specimen holder is biased.

This parameter is used for information purposes only and is not used by MpSys.

Field type: real number Unit: V Default value: 0.0

Integrated charge **integrated_charge**

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: (C Default value: 0.0

Specimen

Origin **origin**

A statement as to where the specimen was obtained from for the experiment.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Composition **composition**

A list of elements and their proportions describing the stoichiometry of the specimen. Each element has a proportion value and is separated by a comma for a given layer and each layer is separated by a semi-colon. Example: Cu 2, Ni 1; Au 1 translates to layer 1 having 2 copper atoms per nickel atom and layer 2 being purely made up of gold.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Scanning

The following parameters affect the ion beam scanning hardware of the MicroDAS system. They cannot be changed once data acquisition has started. See the Beam scanning section in the Data Acquisition chapter for more information about using these parameters.

Enabled **scan.enabled**

A boolean statement as to whether scanning will be used in the next experiment. If the value is set to 'no' then data acquisition will take place without the beam being scanned.

Field type: yes/no Default value: yes

Mode **scan.mode**

Use this parameter to select the shape of the path of the scan.

Permissible values are: "raster", "triangle", "shape"

Field type: restricted text Default value: "raster"

X resolution **scan.x_resolution**

The number of pixels used in producing the x-axis scan.

Field type: positive integer number Unit: pixels Default value: 256

Y resolution **scan.y_resolution**

The number of pixels used in producing the y-axis scan.

Field type: positive integer number Unit: pixels Default value: 256

Scan width **scan.width**

The physical length of the x-axis scan.

Field type: positive real number Unit: (m Default value: 100.0

Scan height **scan.height**

The physical length of the y-axis scan.

Field type: positive real number Unit: (m Default value: 100.0

Scale **scan.scale**

Field type: positive real number Default value: 1.0

Count **scan.count**

The number of complete scans over the specimen. A zero setting permits the scanning to continue indefinitely. Data acquisition automatically stops once the scan count has been reached.

Field type: positive integer number Default value: 0

Dither **scan.dither**

Field type: positive integer number Default value: 1

Interlace **scan.interlace**

Setting this parameter provides a convenient way of getting a fast overview of the specimen when using the "raster" scan mode. A value greater than 1 causes the raster scan to skip scan lines on the first pass. On subsequent passes, the scan fills in the skipped scan lines. A scan is complete when all skipped scan lines are filled in.

This parameter is only used if the scan.mode parameter is set to "raster".

Field type: yes/no Default value: no

Trigger **scan.trigger**

This parameter determines the trigger source use for creating the scan steps. For internal triggering use the "clock" value. To trigger on incoming data use the "data" value and to trigger on incoming charge use the "charge" value.

Permissible values are: "clock", "data", "charge"

Field type: restricted text Default value: "clock"

Clock	**scan.clock**

The dwell time for each scan step when the trigger is set to "clock". A figure less than 1000 can lead to eddy current ghosting when magnetic scanning coils are used.

Field type: positive real number Unit: (s Default value: 1000.0

Events	**scan.events**

The number of external events to receive in the 'Event counter' input at the back of the MicroDAS unit or data events before advancing the scan position. This parameter is only used if the scan trigger setting is "external" or "data".

Field type: positive integer number Default value: 1

Deadtime

Mode	**deadtime.mode**

Permissible values are: "none", "full", "pseudo"

Field type: restricted text Default value: "none"

Timer resolution	**deadtime.timer_resolution**

Field type: positive integer number Default value: 1

Charge resolution	**deadtime.charge_resolution**

Field type: positive integer number Default value: 1

Blanking	**deadtime.blanking**

Field type: yes/no Default value: no

Station Parameters

General

Enabled	**stX.enabled**

A boolean statement as to whether this station is being used for data acquisition.

Field type: yes/no Default value: no

Detector name **stX.detector_name**

The name of the detector associated with this station.

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Detector type **stX.detector_type**

The type of detector used for this station.

Permissible values are: "x-ray", "rbs", "other"

Field type: restricted text Default value: "x-ray"

Radiation detected **stX.radiation_detected**

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Active area **stX.active_area**

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: mm2 Default value: 0.0

Specimen distrance **stX.specimen_distance**

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: mm Default value: 0.0

Energy resolution **stX.energy_resolution**

The energy resolution of the detector used for this station.

MpSys requires this parameter for displaying elements on energy spectrum windows as well as producing element maps.

Field type: positive real number Unit: keV Default value: 0.0

Shaping time **stX.shaping_time**

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: (s Default value: 0.0

Coarse gain **stX.coarse_gain**

The coarse gain setting on a spectrum amplifier module used to amplify the detector signal.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Default value: 0.0

Fine gain	**stX.fine_gain**

The course gain setting on a spectrum amplifier module used to amplify the detector signal.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Default value: 0.0

Bias voltage	**stX.bias_voltage**

The voltage required to bias the detector for this station.

This parameter is used for information purposes only and is not used by MpSys.

Field type: positive real number Unit: V Default value: 0.0

Filter	**stX.filter**

This parameter is used for information purposes only and is not used by MpSys.

Field type: text Default value: blank

Elements	**stX.elements**

The elements that have been selected for displaying in the energy spectrum window for this station.

Field type: text Default value: blank

Calibration

Energy Calibrated	**stX.calibrated_e**
X Calibrated	**stX.calibrated_x**
Y Calibrated	**stX.calibrated_y**

A boolean statement as to whether the spectrum associated with this station has been calibrated or not. MpSys reads this parameter every time it draws or redraws a spectrum window. If the value is "yes" then MpSys uses the parameters below to rescale the x-axis. Warning: manually setting this value to "yes" without properly setting the actual calibration parameters will cause unpredictable rescaling of the x-axis of the spectrum for this station.

Field type: yes/no Default value: no

Energy calibration a	**stX.energy_cal_a**
Energy calibration b	**stX.energy_cal_b**
Energy calibration c	**stX.energy_cal_c**

These are the parameters used for calibrating the energy spectrum for this station.

Field type: real number Default value: 0.0

X calibration a	**stX.x_cal_a**
X calibration b	**stX.x_cal_b**
X calibration c	**stX.x_cal_c**

These are the parameters used for calibrating the X spectrum for this station.

Field type: real number Default value: 0.0

Y calibration a	**stX.y_cal_a**
Y calibration b	**stX.y_cal_b**
Y calibration c	**stX.y_cal_c**

These are the parameters used for calibrating the Y spectrum for this station.

Field type: real number Default value: 0.0

Energy calibration label	**stX.energy_cal_label**

Label used for the x-axis of energy spectrum windows if the calibrated parameter is "yes".

Field type: text Default value: "keV"

X calibration label	**stX.x_cal_label**
Y calibration label	**stX.y_cal_label**

Label used for the x-axis of X and Y windows if the calibrated parameter is "yes".

Field type: text Default value: "Microns"

Normalise on charge	**stX.norm_on_charge**

A boolean statement to determine whether to normalise the y-axis of energy spectra for this station. Normalisation occurs by dividing the y-axis counts by the total integrated charge (also a parameter).

Field type: yes/no Default value: no

6. The Skip Language

Skip stands for Simple Kommand Interface Package.

Skip is a powerful command processing and programming language. It is written to be used as an interface between the user's keyboard and interactive programs.

Beginning users will find the command aliasing and calculator useful. More advanced users and programmers may use the full programming language available. Skip is programmed using a syntax quite similar to 'C' and 'awk'.

Programmers will find that Skip interfaces easily to most pre-existing programs. You can easily add a great deal of power and flexibility to your interactive programs by adding Skip to the "front end" of your command processor.

Command aliasing

You may reduce the amount of typing you do, by defining "aliases" for common command sequences.

Example:

```
alias pr="print"

alias ssave='sort; save; print "Done"'
```

See ALIAS for more information.

Macro execution

You may also place long sequences of commands in a file. Calling this "macro" file will execute the commands as if you had typed them at the command line.

Example:

```
@loadfiles

@printall
```

Macros may be supplied with arguments on the command line. The macros can use these arguments when they are called.

Example:

```
@loadfile "red.data"
```

See MACRO for more information.

Calculator

Skip has a "calculator" built in. The syntax is very similar to the C programming language. You may define variables, assign values to them, and calculate mathematical expressions. Skip provides double precision numeric variables, string variables, and arrays of numbers and strings. You can use the Skip programming language to just do some sums for you, or you can feed values of variables and expressions to your commands. Numeric variables are defined with the "real" command.

Example:

```
real x, y, r

x = 3

y = 4

r = sqrt( x*x + y*y )

print "Plotting circle of radius ", r

plot circle radius=r
```

Input and output redirection

You may redirect the input or output of any command, or sequence of commands, to a file. The syntax is very similar to the UNIX shell input/output redirection. Unfortunately, you must place the input/output redirection at the start of the command. The UNIX shells allow you to put the redirection at the start or the end.

Example:

```
"logfile" print "Loading file ", file[i]

>>"logfile" load file[i]

>>"logfile" print "Done"
```

this could be more easily written by grouping the commands:

```
"logfile" {

print "Loading file ", file[i]

load file[i]print "Done"

                    }
```

Online help

You may obtain help on any of the Skip features or commands by using the "help" command (UNIX users may want to "alias man=help"). Just typing "help" will print some introductory information, then walk you through a series of more detailed topics.

Example:

```
help

help print

help loops

help functions
```

You can obtain help on macros. "help macroname" will print any comments that Skip finds at the start of the macro. So, if you write macros, take the time to put a couple of explanatory comments at the start of your macro files. Comments are started by a "#".

Example:

```
# loadfiles: loads files into the data buffers.

#

#     Usage: "loadfiles filenames"


        for (i = 0; i < $1; i++) {

                load buffer=i file=\$i

        }
```

A programming language

Skip is a full programming language for interactive programs. The Skip programming language is similar to 'C' or 'awk'. Skip provides conditional loops (while, for and do), conditional selection (if and else), compound commands (command list inside curly braces '{ }"), and functions and procedures.

Example:

```
if (x < y) print "x < y"

else print "x >= y"

for (i = 0; i < 10; i++) {
```

```
        print "Loading ", file[i]
        load file[i]
    }

    func sinc( x ) {
        return sin( x ) / x
    }
    plot sinc(x)
```

It will be most useful to use the Skip programming features in macro files and aliases. In this way you may reduce long, repetitive and complex command sequences to a single macro call. See FOR, WHILE, DO and IF in the Command Reference section below for more information.

7. Brief Command Summary

General Commands

Command	Description
Exit	Terminate the MpSys session returning to command prompt
Fnkeys	List or define function key settings
Font	Modify display fonts
List	
Login	
Pointer	Read the current x and y positions of the mouse pointer
Quit	Terminate the MpSys session returning to command prompt
Set	Set an experiment variable
Setopt	Set a system option
warp	Move the cursor to a new x and y screen position

Window commands

Command	Description
Clear	Clear the contents of a window
Kill	Destroy a window
Mapwindows	
New	Create a new window
Redraw	Force a window to redraw itself
Reverse	Reverse the background colour of a window
Setcolor	
Switch	Change the current window to a new window
Text	Display a text string inside a window
Line	Draw a line inside a window
Windows	List brief information about all currently opened windows
winfo	List detailed information about a window

Data Commands

Command	Description
Buffers	Display information about the currently loaded data buffers
Erase	Erase one or more data buffers
Load	Load sorted or unsorted data into a buffer
Mpsort	Sort unsorted data

Data Acquisition

Command	Description
Close	Close a currently running experiment
Run	Run a new experiment
Scan	Start beam scanning
Start	Start data acquisition
Stop	Stop data acquisition
Tune	Tune a map window to update while data is being collected

Spectra Commands

Command	Description
Calibrate	Calibrate a spectrum for a particular dectector
Clmarkers	
Contract	
Down	
Expand	
Extract	
Identify	Identify an element within a calibrated spectrum
Left	
Load	Load a spectra file into memory
Markers	Display current marker positions in a spectrum
Modify	Modify attributes of a spectrum in a plot window
Right	
Save	Save one or more spectra
Show	Display elements in a spectrum
Showrange	
Spectrum	Create a spectrum within a plot window
Sum	Sum all counts within a specified range in a spectrum
Top	
Uncalibrate	Uncalibrate a spectrum
Unzoom	
Up	
Zero	
Zoom	

Map Commands

Command	Description
Clscale	Load a new colour scale into memory
Ctool	Display the colour tool for adjusting map window colours
Load	Load a map file into a map window
Map	Create a map within a map window
Modify	Modify attributes of a map in a map window
Palette	Display the colour palette of the currently loaded colour scale
Save	Save a map to a map file
Shape	Create and manipulate shapes inside a map window

File system commands

Command	Description
Cd	Change the current directory
files	
ls	Directory listing

8. Journal parameter summary

Field types are given by:

T Text

RT Restricted text

I Integer number

R Real number

I+ Positive integer number

R+ Positive real number

YN Yes or No

Default settings of '-' mean that the setting is blank.

General parameters

Name	Description	Set name	Type	Default
Printer Name	Name of printer for dumping windows	printer_name	T	lp1

Experiment parameters

General

Name	Description	Set name	Type	Default
Run ID	Unique identification for current data set	run_id	I+	0
Run date	Date of experiment	run_date	T	-
Run name	Name of experiment	run_name	T	-
Host name	Name of host computer for experiment	host_name	T	-
DAQ name	Name of data acquisition system	daq_name	T	-

Beam line

Name	Description	Set name	Type	Default	Unit
Beam line name	Name of microprobe beam line	beamline_name	T	-	
Object Diaphragm	Diameter of object diaphragm	object_diaphragm	I+	0	(m
Aperture Diaphragm	Diameter of aperture diaphragm	aperture_diaphragm	I+	0	(m
Beam focussed	Whether a focussed ion beam was used	beam_focussed	yesno	yes	

Beam

Name	Description	Set name	Type	Default	Unit
Particle type	Type of beam particle used in experiment	particle_type	RT	H	
Beam energy	Energy of ion beam used in experiment	beam_energy	R+	0	MeV

Chamber

Name	Description	Set name	Type	Default	Unit
Beam current	Ion Beam current detected on the specimen holder	beam_current	R+	0.0	(C
Scattering angle	Angle of RBS detector relative to ion beam	scattering_angle	R	0.0	deg.
Vacuum	Vacuum level in specimen chamber	vacuum	R+	0.0	Torr
Bias voltage	Voltage that specimen holder is biased	bias_voltage	R	0.0	V
Integrated charge		integrated_charge	R+	0.0	(C

Scanning

Name	Description	Set name	Type	Default	Unit
Enabled	Will scanning be used during data acquisition?	scan.enabled	yesno	yes	
Mode	Scan mode: raster, triangle, shape	scan.mode	RT	raster	
X Resolution	Number of pixels producing y-axis scan	scan.x_resolution	I+	256	pixel
Y Resolution	Number of pixels producing x-axis scan	scan.y_resolution	I+	256	pixel
Scan width	Physical length of x-axis scan	scan.width	R+	100.0	(m
Scan height	Physical length of y-axis scan	scan.height	R+	100.0	(m
Scale		scan.scale	R+	1.0	
Count	Number of complete scans over specimen	scan.count	I+	0	
Dither		scan.dither	I+	1	
Interlace	Scan every alternate scan line	scan.interlace	yesno	no	
Trigger	Type of trigger source for creating scans	scan.trigger	RT	clock	
Clock	Dwell time used for "clock" scan mode	scan.clock	R+	1000.0	(s
Events	Number of external events used to advance the scan in "external" or "data" scan modes	scan.events	I+	1	

Deadtime

Name	Description	Set name	Type	Default
Mode	Mode for deadtime detection: full, pseudo or none	deadtime.mode	RT	none
Timer resolution		deadtime.timer_resolution	I+	1
Charge resolution		deadtime.charge_resolution	I+	1
Blanking	Will beam blanking be used when obtaining deadtime information?	deadtime.blanking	yesno	no

Station parameters

General

Name	Description	Set name	Type	Default
Enabled		stX.enabled		
Detector name		stX.detector_name		
Detector type		stX.detector_type		
Radiation detected		stX.radiation_detected		
Active area		stX.active_area		
Specimen distance		stX.specimen_distance		
Energy resolution		stX.energy_resolution		
Shaping time		stX.shaping_time		
Coarse gain		stX.coarse_gain		
Fine gain		stX.fine_gain		
Bias voltage		stX.bias_voltage		
Filter		stX.filter		
Elements		stX.elements		

9. File Formats

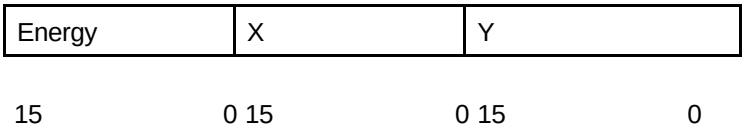
MpSys raw event data (.evt)

The MpSys event-by-event file (extension .evt) is the standard raw data file generated when collecting data. It is a sequentially created file which is made up of event packets that are appended to the file as data is collected.

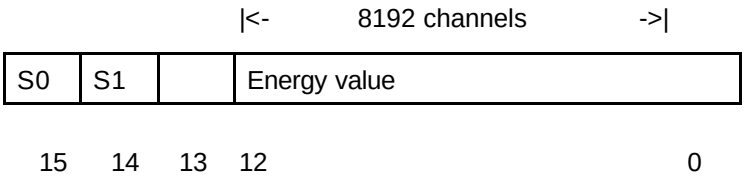
Each event packet consists of three 16-bit data words representing:

- the energy level of the event
- the X scan position at the time the event was collected
- the Y scan position at the time the event was collected

Included within each data word are two data bits describing which of the four data acquisition stations the event was collected with.



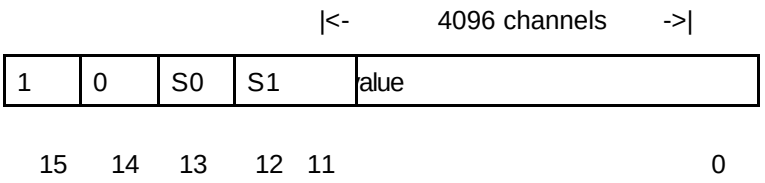
Energy data word



Bits 0–12 make up the actual energy value which comprises a maximum of 8192 channels. Bit 13 is reserved. Bits 14 and 15 describe which station was used in collecting the event and the conversion table is shown below:

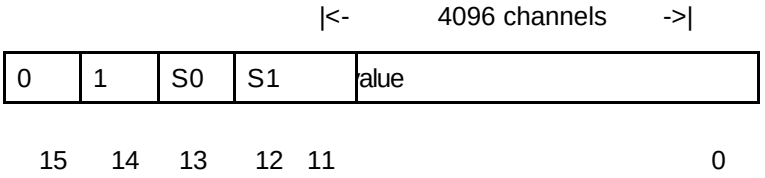
S0	S1	Station #
0	0	1
1	1	2
2	0	3
1	1	4

X data word



Bits 0-11 make up the actual X scan position value which comprises of a maximum of 4096 channels. Bits 12 and 13 describe the station number similarly to how the energy data word describes it above. Bits 14 and 15 identify the data word as an X data word as opposed to a Y data word. This information is used by MpSys for error checking.

Y data word



This word is the similar to the X data word except bits 14 and 15 are swapped.

Determining bad data words

As event packets are collected, MpSys keeps a record of good and bad data words. A bad data word will have incorrect bit settings for the station number or identification. If we let E be the incoming energy data word for a particular event, X be the X scan position data word and Y be the Y scan position data word, then an event is bad if:

E15,14 (X13,12 (Y13,12 or (X15 (1 and X14 (0) or (Y15 (0 and Y14 (1)

Even though bad data words are still recorded in the raw event file, both spectra and map images ignore them.

MpSort can be used to generate sorted data (SD) and sorted pointer (SP) files which filter out bad data words.

MpSys sorted event data (.sd, .sp)

To improve the speed of reconstruction of spectra from collected data, EVT files can be converted into sorted data files (SD) and sorted pointer files (SP). Both files must be in the same directory when they are loaded.

SD files are made up of arrays of events arranged in numerical order (not time order as was the case for the event-by-event file).

SP files consist of n 32-bit words which act as file pointers to each event within the associated SD file.

The size of the SP file will always be:

size = MAX_YCHANNELS * NUM_OF_STATIONS * YPOINTER_SIZE * 4 bytes

Where by default:

MAX_YCHANNELS = 4096

NUM_OF_STATIONS = 4

YPOINTER_SIZE = 2

Therefore size = 128 kilobytes.

The sorted data files are created from the event-by-evet files with the command:

`mpsort <event-by-event file stem>`

MpSys will read sorted data files to produce maps, etc.

MpSys spectra files (.img)

This file contains spectra information for Energy and X and Y beam scan positions for four data collection stations. It consist of 4-byte words representing an event count inside a given Energy, X or Y channel.

The size of the IMG file is:

NUM_STATIONS * NUM_ENERGY_CHANNELS * 4 bytes +

NUM_STATIONS * NUM_X_CHANNELS * 4 bytes +

NUM_STATIONS * NUM_Y_CHANNELS * 4 bytes

= 4 * 8192 * 4 + 4 * 4096 * 4 + 4 * 4096 * 4 = 256 kilobytes

The file is structured as follows:

Energy			X			Y		
Station 1	. . .	Station 4	Station 1	. . .	Station 4	Station 1	. . .	Station 4
0 ... 8192		0 ... 8192	0 ... 4096		0 ... 4096	0 ... 4096		0 ... 4096

MpSys 2.X allowed use of simplified IMG files which had only enough spectrum information for one station as well as for one of Energy, X or Y. MpSys 3 encourages use of the complete IMG file for all four stations as there is no means of telling within a simplified IMG file exactly what station the spectrum was collected on.

MpSys Map files (.map)

Mpsys MAP files are binary data files containing energy intensity information over a scanned region.

The structure of the MAP file is shown below:

X	X	W	H	DATA

Bytes: 1 1 2 2 W * H * 4

X = Future expansion (unused in MpSys 2.X)

W = Map width (2 bytes)

H = Map height (2 bytes)

Each map intensity datum is 4 bytes.

10. MpSys X-ray line database

This matrix shows the number of x-ray lines in the MpSys data base for each element.

Name	K	L	M	Name	K	L	M	Name	K	L	M	Name	K	L	M
H				Zn	2			Pr		13		Ra		15	
He				Ga	2			Nd		13		Ac		4	
Li				Ge	2			Pm		5		Th		15	6
Be				As	2			Sm		13		Pa		15	6
B				Se	2			Eu		13		U		15	6
C	1			Br	3			Gd		13					
N	1			Kr	3			Tb		13					
O	1			Rb	3			Dy		13					
F	1			Sr	3	9		Ho		13					
Ne	1			Y	3	9		Er		13					
Na	1			Zr	3	11		Tm		13					
Mg	1			Nb	3	11		Yb		13					
Al	1			Mo	3	11		Lu		13					
Si	2			Tc	3			Hf		13	2				
P	2			Ru	3	11		Ta		13	2				
S	2			Rh	3	11		W		13	2				
Cl	2			Pd	3	11		Re		15	2				
Ar	2			Ag	3	12		Os		15	2				
K	2			Cd	3	11		Ir		15	2				
Ca	2			In	3	11		Pt		15	4				
Sc	2			Sn	3	12		Au		15	4				
Ti	2			Sb	3	12		Hg		15	3				
V	2			Te	5	12		Tl		15	4				
Cr	2			I	3	12		Bi		15	6				
Mn	2			Xe	3	1		Po		11					
Fe	2			Cs	6	13		At		4					
Co	2			Ba	3	13		Fr		5					
Ni	2			La		13		Rn		4					
Cu	2			Ce		13		Ra		15					

11. Working with MpSys in a Microsoft Windows environment

(Based on a MARCO Technical Report - October 1998)

Many of our collaborators ask us if a PC version of the MpSys data acquisition system is available.

Surprisingly, the answer is yes!

This section explains how this is done. First a little history. Over the past 15 years, the MARC laboratory employed a wide variety of different computers to perform data acquisition and analysis. The concept of Total Quantitative Scanning Analysis (TQSA), pioneered by George Legge, was implemented on each computer in a program we always called MpSys. This was not always an easy task. With each generation of computer, the task never became easier as demands for more graphics intensive facilities became ever greater.

To meet the demands of TQSA, specialised hardware and software had to be developed. The previous generation of data acquisition system illustrates one solution. The system consisted of:

- a dedicated VME crate, with its own computer
- a custom designed and built Fast Data Acquisition Crate (FDAC)
- a dedicated workstation to display the data

This provided the necessary computer power to collect data from up to four detectors simultaneously, while still allowing for FAST specimen scanning with the advantage of a random scan pattern. The system was sophisticated, expensive to make and maintain, but allowed count rates of up to about 5kHz per detector.

The advent of fast and inexpensive PC's has allowed all this complexity to be eliminated. The latest generation of our TQSA data acquisition system, the fastest and best yet, is also the simplest. It consists of:

- a dedicated PC with a National Instruments LabPC card
- a simple interface unit called MicroDas

The PC has sufficient power to simultaneously collect, store and display the data from up to four detectors. This system is also extremely robust, it supports data acquisition rates in excess of 20kHz per detector. The key to the new system is the linux operating system. This allows for extremely fast and efficient low level code to service the data acquisition needs. Linux offers overwhelming advantages for time critical tasks like data acquisition. These include:

- the ability of insert efficient low level code into the kernel
- security in a multi-user environment
- extreme robustness: no "blue screen of death" or "general protection fault" common under Windows.

After two years experience with the new system, and sales to several other laboratories worldwide, the new system has shown its qualities very well.

But still some users prefer a Windows environment instead of the less user-friendly unix environment. There can be no doubt that Windows offers many advantages (except for data acquisition!). Fortunately MicroDas can be seamlessly integrated with a Windows environment. This is accomplished by using a second, Windows-equipped, computer to control MicroDas. All the second computer requires is a X-server program, such as the excellent DEC product: eXcursion, running under Win3.1, Win95, Win98, WinNT, or later.

An icon on the Windows machine desktop invokes MpSys on the linux machine and pops up the MpSys graphical user interface. The data disk on the unix machine is mapped to a network drive on the Windows machine. The user does not even know they are running a process on a unix machine. All functions of the power of the unix system are provided with the convenience of the Windows environment!

With this way of running MpSys, the user has the convenience of being able to control the data acquisition system from ANY suitably equipped Windows PC. In the laboratory, in the office, at home, overseas.... Also, advanced users can still unleash the power of unix for their projects.

The total cost of all the hardware and software required for the complete system is less than the cost of the VME crate alone in the previous generation data acquisition system!

End of Manual.